

国际信息工程先进技术译丛

Springer

基于视觉的自主机器人导航

Vision Based Autonomous Robot Navigation: Algorithms and Implementations

Amitava Chatterjee
(印度) Anjan Rakshit 编著
N. Nirmal Singh
连晓峰 等译





机械工业出版社
CHINA MACHINE PRESS

国际信息工程先进技术译丛

基于视觉的自主机器人导航

Amitava Chatterjee
(印度) Anjan Rakshit 编著
N. Nirmal Singh
连晓峰 等译



机 械 工 业 出 版 社

移动机器人自主导航是机器人领域的热点研究内容。本书首先介绍了基于机器视觉感知机制的自主移动机器人导航的理论与研究，重点介绍了基于地图、地图构建以及无地图环境下的基于视觉传感器的机器人导航算法，并给出了实际应用中基于视觉的机器人先进导航算法的实现。本书详细阐述了实际应用中基于视觉的自主导航算法以及SLAM问题，提出了利用视觉感知如何实现子目标驱动的导航概念，提出了利用模糊逻辑进行基于视觉的机器人路径跟踪概念，以及如何利用基于微型控制器的传感器系统研制实验室环境下的低成本机器人。

本书可作为从事机器人研究，尤其是移动机器人方面的研究人员的参考书，也可作为高等院校自动化、计算机等相关专业研究生以及教师的参考用书。

本书版权登记号：图字：01-2013-6383 号

图书在版编目（CIP）数据

基于视觉的自主机器人导航/（印）查特吉（Chatterjee，A.），（印）莱克舒（Rakshit，A.），（印）辛格（Singh，N. N.）编著；连晓峰等译．—北京：机械工业出版社，2014.2

（国际信息工程先进技术译丛）

书名原文：Vision based autonomous robot navigation：algorithms and implementations

ISBN 978-7-111-45674-2

Ⅰ.①基… Ⅱ.①查…②莱…③辛…④连… Ⅲ.①机器人视觉－视觉导航 Ⅳ.①TP242.6

中国版本图书馆 CIP 数据核字（2014）第 022876 号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
策划编辑：顾 谦 责任编辑：顾 谦
版式设计：霍永明 责任校对：刘怡丹
封面设计：马精明 责任印制：乔 宇
北京机工印刷厂印刷（三河市南杨庄国丰装订厂装订）
2014年3月第1版第1次印刷
169mm×239mm · 13.5 印张 · 258 千字
0 001—3 000 册
标准书号：ISBN 978-7-111-45674-2
定价：59.90 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
社服务中心：(010)88361066	教材网：http://www.cmpedu.com
销售一部：(010)68326294	机工官网：http://www.cmpbook.com
销售二部：(010)88379649	机工官博：http://weibo.com/cmp1952
读者购书热线：(010)88379203	**封面无防伪标均为盗版**

译　者　序

移动机器人自主导航是机器人领域的热点研究内容。世界范围内开展了大量该领域的广泛研究，且已得出一些非常有价值和应用前景的研究成果，这对于移动机器人在工业、社会服务业、空间探索等多个领域有着重要意义。

本书首先介绍了基于机器视觉感知机制的自主移动机器人导航的理论与研究。重点全面，综合介绍了基于地图、地图构建以及无地图环境下的基于视觉传感器的机器人导航算法，并给出了实际应用中基于视觉的机器人先进导航算法的实现。

本书是专门研究基于机器视觉感知的自主移动机器人导航理论的著作。传统的机器人导航系统主要利用声纳、红外、GPS（全球定位系统）以及激光传感器等，这些方法都由于传感器本身特性或成本等限制而存在一定程度的不足。而视觉感知已逐步代替了上述传感器，可降低系统成本，同时提高系统的智能性、灵活性和鲁棒性。

本书详细阐述了实际应用中基于视觉的自主导航算法以及 SLAM 问题，提出了利用视觉感知如何实现子目标驱动的导航概念，同时提出了利用模糊逻辑进行基于视觉的机器人路径跟踪概念，以及如何利用基于微型控制器的传感器系统研制实验室环境下的低成本机器人。书中还介绍了利用现有机器人与外设集成实现低成本机器人的成功案例。

本书的最大亮点在于，详细介绍了在 32 位 Windows 环境下如何具体实现基于视觉的导航模块，另外还讨论了如何采用两个摄像头实现基于视觉的 SLAM 系统概念。

本书主要由连晓峰翻译、审校、整理，并对原书中的错误进行了注释，王佩荣、贾琦、李东红、潘媛、王炜伊、郭柯、王宇龙、张晓伟、闫峰、金成学、张子康、王伟等人也参与了部分翻译工作。

本书可作为从事机器人研究，尤其是移动机器人方面的研究人员的参考书，也可作为高等院校自动化、计算机等相关专业研究生及教师的参考用书。

限于译者的经验和水平，书中难免存在缺点和错误，敬请广大读者批评指正。

译　者

原书前言

本书致力于采用基于计算机视觉的感知机制进行移动机器人自主导航的理论与开发。传统的机器人导航系统利用如声纳、红外、GPS（全球定位系统）、激光等传统传感器，而这些传感器具有如物理局限性或成本高等一些缺点。视觉感知作为一种常用的替代方式，可减少整体成本，维持高度智能性、灵活性以及鲁棒性。

第1章详细阐述了移动机器人自主导航的基本概念，并重点介绍了采用视觉作为感知机制。在此，对室内、室外环境下基于视觉的导航研究工作进行了大致分类。之后，介绍了不同形式的障碍物检测和避障方法。第2章讨论了如何成功实现外围设备与现有移动机器人的实际接口，在此详细介绍了通过以中断驱动模式的串口通信实现KOALA机器人与外设的接口。第3章详细介绍了一种基于视觉的机器人导航策略，其中采用一个基于子目标的机制来沿最短路径达到最终目标，同时还实现期望的避障。该策略采用一个两层架构，其中视觉传感器工作于第一层，而基于红外传感器的避障机制工作在第二层。

第4章讨论了如何在实验室开发一个具有特殊功能的低成本机器人，重点是开发两个基于微控制器的机器人传感器系统，包括：开发一个具有动态距离增强功能的红外测距系统；采用开关模式同步检测技术开发一个光学接近检测系统。在第5章，一步步地从简单模块逐步发展为复杂模块，讨论了如何在32位Windows环境下实际实现一个基于视觉的导航子程序。

在第6章和第7章讨论了在移动机器人导航中集成模糊逻辑的问题。其中，首先讨论了如何开发一个室内路径跟踪的基于视觉的导航机制，在此基于视觉的模糊导航与基于红外的模糊避障机制相结合。第7章首先介绍了移动机器人基于EKF的SLAM概念，然后讨论了一个更复杂的场景，其中在具有不正确或不确定的传感器统计信息情况下，可有效利用模糊或模糊神经网络监督来提高基于EKF的SLAM性能。第8章讨论了如何实际实现一个基于双摄像头的视觉系统以实现室内环境下的SLAM。

加尔各答，西孟加拉邦，印度

Amitava Chatterjee

Anjan Rakshit

N. Nirmal Singh

2012年9月

目　　录

第1章　移动机器人导航

摘要：本章介绍了移动机器人自主导航的基本概念，以及利用视觉传感器作为感知机制以实现预期目标的实用性。本章还讨论了在室内和室外环境下基于视觉的导航分类，并且还介绍了在该研究领域下各种显著的研究方向，以及多种不同的障碍物检测和避障方式。

1.1　自主移动机器人导航

近年来，机器人领域的一些先进技术的发展进步已对许多工业生产和社会发展作出了巨大贡献。如今，机器人系统已在工厂自动化生产、监控系统、质量控制系统、AGV（自主导引车）、抗灾和医疗辅助等领域得到广泛应用。越来越多的机器人应用正在改善我们的日常生活，同时在执行各种不同任务方面，机器人也前所未有地得到了更多关注[1]。对于很多这样的应用，机器人的自主移动性是一个非常关键的问题[100]。自主移动机器人是指在没有人工导引的结构化或非结构化环境中能完成预期任务的机器人。一个完全自主的移动机器人应具备以下能力：

1）能够获取环境的相关信息；

2）能够在没有人为干预的情况下长时间工作；

3）无需人工辅助能够在整个工作环境中整体或部分运动；

4）除非是机器人的设计规范，否则能够避免出现危害人类、财产或其自身的情况。

一个自主移动机器人还应能够学习或获取新的能力，如调整策略以完成任务或自适应环境的变化。

1.2　为何要视觉导航

视觉是能够让我们人类获取客观世界相关信息的一种感知，确切地说，这也是人类最依赖的一种感知。近年来，能够提取上述信息的计算机视觉技术已得到不断地发展和完善。虽然视觉信息处理的计算量庞大，但随着处理速度更快且价格更低的处理器不断地发展，对于各种复杂任务，基于视觉的移动机器

人实时导航系统正逐渐成为现实，并且在该研究领域，也前所未有地吸引了更多的研究工作[100]。

用于机器人导航的其他传感器包括红外传感器、声纳传感器、激光测距仪、位置感知装置(PSD)以及惯性传感器等。红外传感器的应用非常有限，常用作接近检测器，作为测距仪的主要缺点在于测距范围有限且易受环境光的干扰，另外还具有非线性特性且反射率取决于目标物体的表面[2]。声纳传感器计算量较小且易于读数，但测量数据由于环境干扰而可靠性较低。声纳测距仪可测量与物体之间的距离，但由于其波束较宽而导致角度分辨率较差[3]。与声纳传感器相比，激光测距仪可提供更高可靠性的瞬时测量、更精确的测距精度和角分辨率以及更佳的方向分辨率，但成本也更高。激光传感器还可提取除了距离之外的信息，比如通常利用激光扫描仪的功能来提取拓扑信息，以确定物体表面的纹理结构以及精确的距离信息。激光测距仪的一个缺点是激光扫描可能会检测不到透明物体(如玻璃和窗户)。惯性导航传感器(如加速度计和陀螺仪)可提供运动车辆的朝向和轨迹测量，但无法提供车辆运行环境中的障碍物信息。GPS(全球定位系统)是目前在导航系统中最常用的辅助工具之一，GPS 可提供绝对或相对的实时位置数据，但与接近操作的一般需求相比，其精确性和带宽有限。GPS 的性能可能会受到卫星“视线”遮挡的影响，从而导致其精确性和更新率下降[4]。这些测距传感器都难以检测地面上小型或扁平的物体，也无法区别不同类型的地表。尽管这些小目标和不同类型的地表难以通过测距传感器进行检测，但大多数情况下都可由一个被动传感器(如摄像头)来检测。一个视觉系统可看作一种被动传感器，与看作主动传感器的红外、激光和声纳等传感器相比，其具有一些基本优势[5]。被动传感器(如摄像头)通过发射光或波来获取数据而不会改变环境，并且所获得的图像中包含更多的信息(如实质性、空间和时间信息)。因此，上述所有传感器所获取的物理环境相关信息都要少于摄像头所获得的信息。同时，随着更快、更廉价的计算能力的不断提高，正逐渐设计出基于视觉的真实世界导航系统。摄像头价格低廉，即使是最贵的摄像头也相对经济、实惠。因此，视觉作为移动机器人的一种感知方法提供了非常具有吸引力的解决方案。

1.3 基于视觉的导航

基于视觉的机器人导航是一种主要利用视觉传感器来引导移动机器人避开静态(或可能是动态)障碍物到达预期目的地或在环境中沿一条预期路径运动的技术。一般来说，基于视觉的机器人具有一个能够感知外界环境的视觉系统。通常，一个基于视觉的自主机器人在视觉系统中具有五个主要组成部分：

1. 地图

系统需要外界环境的一些内部表征或知识以完成目标驱动的任务。

2. 数据获取

系统通过摄像头来采集图像。

3. 特征提取

从采集的输入图像中提取显著特征，如边缘、纹理和颜色。

4. 路标识别

根据某些预设标准，系统在观测图像中的特征和内存中预存的预期路标之间寻找可能存在的匹配信息。

5. 自定位

在自定位阶段，计算作为检测路标和之前位置的一个函数的机器人当前位置，然后系统推导出机器人的运动路径。机器人的运动过程可仅仅进行避障和/或是以目标驱动。

移动机器人的导航问题大多数情况下可分为以下四个子问题[7]：

1. 环境感知

感知环境并将其抽象为一系列特征。

2. 路径规划

利用特征来构建机器人应达到的一系列目标点。

3. 路径生成

接下来，通过一系列目标点来获得一条路径。

4. 路径跟踪

利用控制器来保证移动机器人沿预期路径运动。

一般来说，开发一个能够检测并避开障碍的基于视觉的机器人系统是相当复杂的，这是由于从包括机器人和障碍物的现场视频流中提取信息，并利用尽可能少的计算处理来实现预期实时性能是一项非常复杂的任务。在过去的 20 年中，未知环境下机器人运动的问题得到了广泛关注。机器人应能够以智能方式避开可能遇到的各种不同类型的障碍物。相应地，大量研究工作致力于利用计算机视觉实现通过对感知数据进行逻辑分析来实现避障的具有导航能力的基于视觉的自主移动机器人系统。大多数这些研究的主要目标是定位静态或动态障碍物，从而可规划出一条合适的路径以使得机器人绕开障碍物，并最终按所规划的路径运动。利用视觉感知的室内和室外导航已成为移动机器人领域中两个主要的研究方向。

1.3.1 基于视觉的室内导航

基于视觉的室内导航机制大致可分为三类：

1）基于地图的导航；

2）基于地图构建的导航；

3）无地图导航[6]。

1.3.1.1 基于地图的导航

在基于地图的导航过程中，系统已具有环境的先验知识，并利用地图信息进行导航。这些环境地图的形式主要有几何模型、拓扑地图或图像序列[8,9,10]。在早期利用地图的方法中将环境细化为不同程度，分别为"占据栅格地图"、"虚拟力场"或"S地图"等形式[11]。这些方法非常容易受到感知误差的影响，在参考文献[12]中对该问题进行了详细阐述。其中，一个重要假设是应对传感器测量的不确定性考虑一定容差[12]。这些方法都称为"绝对定位"，之后对这些研究问题进行改进，采用"增量定位"来解决。在此，假设已知机器人位置的似然信息，在导航过程中通过利用视觉工具的观察不断修正这些信息，并在随后的导航中采取必要措施。FINALE[13]系统正是基于上述概念，其中当某个路标与一个图像特征匹配时，利用空间的几何表示以及机器人位置的统计模型不确定性，采用卡尔曼滤波方法来更新机器人位置的均值和协方差矩阵。另一类方法是基于空间的拓扑表示，其中如在NEURO-NAV[14]中采用了一组神经网络，或如在称为FUZZY-NAV[15]的系统中采用比NEURO-NAV更复杂的监督式模糊控制器。然而，基于地图的导航方法的缺点在于不易产生环境的模型或地图(尤其是米制地图)。

1.3.1.2 基于地图构建的导航

基于地图构建的导航处理程序试图解决没有环境先验信息的机器人开始运动的问题。机器人首先探索周围环境并建立一个内部描述，之后再利用该内部描述进行导航任务。在该研究领域中，大部分都是采用基于空间拓扑表征的方法，参见参考文献[16]，[17]，并研究如何在基于图的空间描述中构建节点、如何区分几个相邻节点、如何考虑传感器不确定性影响等一系列问题。然而，这些方法的主要缺点是难以识别之前访问过的节点。在该领域的其他方法还包括基于占据栅格的表示[18,19]，采用全景视觉的方法[20]，以及综合占据栅格方法和拓扑结构方法优点的方法[21]。文献中还包括一些其他类型的基于地图构建的导航系统，如基于可视化声纳的系统[22]或基于局部地图的系统[23]。这些系统都是在导航过程中采集环境数据并构建一个用于支持在线安全导航的局部地图。该局部地图包括通常为摄像头视场角函数的部分环境中具体障碍物信息和自由空间数据信息。

1.3.1.3 无地图导航

无地图导航方法属于一类更有前景的方法，其中导航过程在没有任何地图的条件下开始进行。这种类型的导航过程也称为反应式导航，这是通过观测、特征/路标识别(通常是墙、桌子、楼道、墙角等自然物体)以及特征跟踪来在线

提取环境的重要信息，然后利用导航算法对所提取的相关有用信息做出“反应”。在这些技术中，一些传统的常用方法是采用光流法[24]和基于外观的技术[25,26]。光流法是模仿蜜蜂的可视化行为，根据机器人左右摄像头获取图像之间的速率差来确定运动。机器人会朝图像变化速率较小的那一侧运动。参考文献[24]在该基本方法的基础上进一步改进，开发了一种利用深度信息的基于光流的导航系统，并且还开发了一个具有俯仰、倾斜、左右幅度控制的更加复杂的立体头部系统。一些研究人员已证明能够获得准确深度信息的立体视觉与光流分析方法相结合可具有更好的导航效果[27,28]。在参考文献[29]中，利用立体信息与来自立体图像之一的光流相结合来构建一个占据栅格地图，并对一个地面机器人执行实时导航策略。另一方面，基于外观的方法通过保存一系列图像来增强环境印象，这一系列图像通常是从向下采样的摄像头采集图像所提取的子窗口中创建的，然后在任意给定时刻，对采集图像与所有模板进行扫描来确定是否与其中某个模板相匹配。如果图像匹配，则在导航中采取相应的控制动作。在此，关键问题是如何提高在训练阶段的图像保存方法，以及随后的图像匹配过程。不利用地图进行环境识别主要有两种方法[30]：

1）基于模型的方法。利用预先定义的对象模型来识别复杂环境中的特征并在环境中自定位。

2）基于视图的方法。该方法并不从预存的图像中提取特征，而是采用图像匹配算法来实现自定位。

1.3.2 基于视觉的室外导航

这些系统无需对导航空间进行明确表示，而是通过识别在环境中所观测到的物体或通过产生基于视觉观测的运动来跟踪这些物体。根据环境结构的程度，室外导航可分为两大类：结构化环境下的室外导航和非结构化环境下的室外导航。在许多情况下，室内导航所采用的地图表示方式并不适用于室外导航，这是由于室外通常都是包含大量物理区域的大型场景，由此会导致表征室外环境的信息量急剧增大到无法估量的地步。在结构化环境中的室外导航主要是指道路跟踪，即检测路线并持续导航的能力。相对于结构化的室内空间，室外环境大多数情况下都是由砂石、花园、人行道和街道组成的。大多数这些元素都具有不同的颜色和纹理，利用这些特征可以很方便地进行室外导航。确定导航区域的第一步就是根据视觉信息对地形进行分类。迄今为止，在道路跟踪方面所报道的最突出的研究工作是NAVLAB工程[31,32]。NAVLAB道路跟踪算法包括三个阶段：第一阶段，对每个道路和非道路像素都采用高斯分布来进行颜色和纹理像素的综合分类；第二阶段，对道路像素执行霍夫变换和投票表决程序，来获得道路消失点和朝向参数；最后，根据确定的道路边缘对像素再次分类。在下一幅图像中重复执行上述分类

过程以使得系统可适应路况的变化。对于结构化环境下的道路检测和跟踪，许多研究工作都是根据上述相关概念进行的[33-36]。在室外环境中采用视觉信息和GPS数据是提高位置估计可靠性的另一种方法[37]。在最近的研究工作中，研究人员还提出了一种特征跟踪与3D立体环境重构相结合的方法。在参考文献[38]中，对于非结构化室外环境，采用立体视觉作为一种新的导航策略。该系统是基于一种更快且更精确的角点检测新方法。在该方法中，3D定位所检测的特征并采用归一化均方差和相关测量来进行特征跟踪。

1.4 研究现状

至今为止，在基于视觉的移动机器人领域已进行了大量的研究工作，且大多数研究都集中于障碍物检测、环境地图构建、安全路径规划和楼道导航。本节将对一些已采用基于视觉导航的不同导航方法进行概述，即基于摄像头采集的视觉信息，且已知或未知环境的先验信息。

"Stanford Cart"是一项众所周知的基于视觉的移动机器人项目[39]，该项目已完成了很长一段时间。该系统采用一个安装在移动平台上的摄像头，将视频信号发送到远程计算机，然后通过这些视频信号来处理和控制机器人的运动，同时还具有一个用于避障和路径确认的规划器。一旦接收到包含环境完整场景的图像，系统就会进行处理。据报道，该处理时间长达1~2h。一旦完成图像处理，就会产生一条规划路径来引导机器人实现避障。该系统的主要问题是信息处理的时间太长。例如，阴影移动会导致机器人在所构建的地图中产生误差。后来，通过图像分割与兴趣算子相结合的思想来检测环境中的不同特征，从而解决上述缺点[40]。

在20世纪90年代早期，Horswill研制开发了一个称为POLLY的机器人[41,42]，该机器人采用单目视觉进行导航，并工作在一个颜色固定的环境中。在导航过程中，POLLY采用一个由用于自定位的一系列路标所组成的拓扑地图，这些路标是在办公室环境中特定位置拍摄的一组照片。为确定机器人当前所处的位置，可通过将每个路标与实时视频数据相匹配来进行路标或位置识别。在该研究工作中，实际上是采用一种等效于人工势场法的局部导航策略。因此，机器人很容易陷入局部最小值。这项工作中所采用的方法主要可分为四步：平滑图像；从前景的梯形区域中确定平均像素值；将该平均像素值作为地板，然后从图像底部开始标记每个像素，并对每列进行扫描，直到出现错误匹配；列的高度表明了机器人与障碍物之间的距离，即径向深度地图。这种方法的主要问题在于经常会将阴影误判为障碍物，这主要也是由于机器人视觉处理系统过于简单所造成的。这种处理过程基于在图像中简单提取无纹理的地板，从而确

定可运动到的自由空间。在以后的几年中，研究人员还提出只具有一个摄像头的移动机器人导航机制[43,44]。同时还提出采用占据栅格的地图构建框架和特征位置检测算法，即通过单个摄像头在线处理 RGB 彩色图像序列[45]。该系统并不是采用匹配方法，而是计算每个位置处找到物体的概率。该算法首先在当前图像帧中通过 Harris 边缘和角点检测器来检测物体的轮廓边缘。在图像序列的开始处，在占据栅格地图中增加边缘特征，对该边缘特征扫描可确定峰值。然后，考虑任意深度下的所有可能位置，将检测到的特征投影到 2D 图像平面。根据里程计数据和提取的图像特征，系统定位模块可计算机器人位置。利用之前图像中边缘和特征的颜色或梯度有助于增强在某一特定位置存在物体的可信度。室内环境下实验的网格大小设置为 25mm × 25mm。在连续图像中，允许机器人移动 100mm。由于采用单个摄像头只能获取前向信息，但这些信息对于室内导航已足够，且性价比较高。对于室内未知环境，采用单摄像头的其他研究工作大多集中于采用中线跟踪和沿墙跟踪的方法[46]。最近，参考文献[47]中提出了一种采用单个 USB 摄像头和笔记本电脑实现的轮式移动机器人走廊中线跟踪方法。在该实验中，采集的图像大小为 320 × 240 像素，前视场大小为底部 60cm、顶部 20m，且机器人运动速度为 0.823km/h。该方法也采用霍夫变换来检测走廊和墙壁的边界。当走廊中没有障碍物时，机器人将沿走廊中线运动。在此所用的障碍物检测方法是改进的 Ulrich 法[95]。这种改进是忽略了由于光线影响导致的障碍物错误检测。在参考文献[47]中，一旦检测到任何障碍物，则根据障碍物大小、与机器人的距离以及走廊宽度来决定是避障还是停止，这些都是由真实空间中的 2D 位置和图像中的任意位置来确定的。另外还有研究人员提出了一种基于定性的方法，即利用一个现成的未标定的前向观测摄像头进行室内和室外导航[1]。在此，采用一种基于示教—重现的方法。在示教阶段，用户手动引导机器人沿着所规划的路径运动。这时，机器人采用 Kanade-Lucas-Tomasi(KLT)特征跟踪器[48]来选择和跟踪特征点，并保存在数据库中。在重现阶段，应建立当前图像中特征点的坐标与示教阶段在第一幅图像中所选择特征点的坐标之间的对应关系，由此来确定转弯命令。同理，在立体视觉中也开发了一种类似的基于用户经验的示教—重现导航算法[49]。

在立体视觉中，机器人可通过两个或多个摄像头来测量距离信息，也可采用声纳传感器。然而，使用两个或多个摄像头会增大处理成本。据报道，目前在一些移动机器人导航研究工作中已采用两个或多个摄像头[50-54]。利用两个摄像头为获得深度信息，必须已知摄像头和机器人头部的相关几何信息。在立体视觉中要获得深度信息，要求两个摄像头的视线需在场景中的 P 点相交，在该点处进行处理以获得深度信息。基于立体摄像头的系统在导航中可用于特征识别、特征跟踪以及实时计算 3D 特征点的距离。这很大程度上取决于在一帧图像

中图像点/特征的选择/提取，以及随后在不同位置和/或不同朝向上采集的相同场景连续图像帧中对这些点或特征的跟踪。除非能够识别并跟踪明显特征，否则基于特征的视觉系统无法工作[55]。获取图像特征的3D坐标非常重要，这便于计算像平面上所选特征与机器人的摄像头焦点之间的距离。这些3D坐标对于机器人定位和场景解释十分重要。近年来，已提出一些利用鱼眼立体视觉[56]、单目视觉[57]、视觉与声纳传感器相结合[58]以及从特征检测器(FD)接收预处理输入的生物启发显著图(SM)等来测量特征/物体与摄像头之间距离的方法。在参考文献[59]中，两个摄像头的FD和SM之间的相互作用可用于左右两幅图像中相应路标的检测，并可估计路标方向和距离。一个神经网络将七个给定标识符（所有图像中路标的X和Y坐标、两个摄像头的平移角度和一个共同的摄像头倾斜角)与路标的方向和距离相对应。在LAGR机器人中提出了一种基于立体视觉的栅格地图构建方法来进行室外导航[60]。除了视觉传感器，还集成了惯性导航单元、GPS接收器和前端防碰开关来进行外界感知。其中，地图构建方法主要包括以下五个步骤：

1）用一对彩色摄像头来采集图像；

2）根据摄像头的相对几何关系，在两幅图像的匹配块中确定3D表征；

3）转换每个坐标点来确定机器人的瞬时俯仰角和横滚角，从局部坐标系转换到全局坐标系所估计的机器人惯性导航和偏航角；

4）对地形图进行求导操作来寻找斜率上的突变；

5）利用包括地形及其导数估计的新测量值来更新全局地图。

近年来，成功开发了一种基于立体视觉的算法以用于NASA(美国国家航空航天局)的火星探索漫游机器人Mars Exploration Rover在具有潜在危险的地形下进行自主导航。该项目中的具体操作步骤包括：对摄像头采集图像下采样256×256像素以减少计算时间；然后通过将第一幅图像中物体的外极线水平投影到另一幅图像上的同一物体来对图像对进行处理；接下来计算两幅图像的拉普拉斯算子，并与视差范围内选择的潜在匹配相关联，对图像中的所有像素重复执行上述过程，从而使得如果这些像素的估计值匹配失败，则舍弃；最后每个视差值都可利用摄像头几何模型映射到一个3D表示。

在开发基于视觉的机器人导航策略过程中，采用全方位视觉也是一个非常重要的方式。全方位摄像头可对处于任何位置的机器人提供图像形式的360°环境场景。这种视觉系统通常采用鱼眼摄像头和全景摄像头。优点是可处理全视场图像，但随之而产生的缺点是由于所选摄像头特定类型的几何尺寸，导致成本和视觉算法开发复杂性都比较高。目前已具有一些基于全方位视觉的导航系统[62-65]，其中大多数都是采用基于光流的导航技术。

光流是指对场景表面相对于摄像头的运动而引发的视觉运动的一种测量。从

计算角度而言，最常见的光流表示是图像空间的2D矢量场，其中每个矢量都描述了一幅图像中某点到下一幅图像中相应位置的运动。在一个视觉导航的机器人中，光流很大程度上是由机器人运动而产生的摄像头运动来推算的。对于光流法，许多研究工作都是基于如前所述的蜜蜂飞行时观察所采用的路径中心跟踪方法。据观察，蜜蜂的飞行方向和速度都与其相对于环境运动所导致的视觉运动直接相关[66,67]。基于光流，移动机器人在走廊中的导航就是通过在不同位置放置的不同类型摄像头来实现的。例如，在参考文献[68]中，采用了一个前向主动的广角摄像头来实现走廊中心定位。在此，利用图像左右2/3边缘来计算光流。通过在左右2/3边缘处维持最大光流，使得机器人位于墙壁和障碍物之间的中心路径上。在此，利用摄像头的观测方向来计算机器人在朝向调整过程中的旋转角度。在参考文献[69, 70]中，采用不同的方法来计算从位于机器人两侧朝向90°的两个摄像头得到的平均横向光流之差。与参考文献[68]不同，这些方法对旋转光流没有进行补偿。实际上，对转向的限制可使得所产生的旋转效果最小。为应对缺乏纹理的情况(如门口或没有墙壁)，采用能够与一侧墙壁之间保持恒定距离的单侧保持行为，这样仍能够提供足够的纹理来估计光流。

采用基于相关技术的光流法在本质上类似于立体视觉中的视差匹配方法。不同之处在于光流法中图像是时间上独立的，而立体视觉中图像是空间独立的。但遗憾的是，光流法中图像关联的任务会由于机器人在真实环境中可能受到振动而变得十分复杂。这意味着与立体视觉不同，不能只局限于在同一水平扫描线上搜索相应图像块[71,72]。另外，还提出了采用基于梯度的方法来实现机器人连续运动下导航的其他光流技术[73-75]。

由于光流提供了在单幅低质量图像中无法细致提取的场景质量特征，目前已证明基于光流的视觉导航技术对于无人机特别有效。在该研究架构下，针对视在运动的处理和应用，已在仿生行为方面进行了大量研究工作。无人机具有一个由感光阵列构成的摄像头眼，其中每个感光器都与一个电子基本运动检测器(EMD)相连，该EMD可计算特定位置上的局部光流[76]。与确定是否存在障碍物的光流计算相比，这种产生光流测量变化的EMD极坐标确定可构建一个包含障碍物位置的局部地图。在参考文献[77]中，也成功实现了一个具有由感光阵列及其相应EMD组成的摄像头眼的无人机。EMD所获得的信息可用于确定是否存在障碍物。另外，当无人机以恒定的速度和高度飞行时，可根据人工视网膜速率建模方程来计算一个参考光流分布。随着地形的变化，系统可通过改变推进力和飞行舵位置来根据参考光流调整在线计算的光流。

巡线导航是另一种在工业生产中广泛应用的基于路标的导航。巡线导航可看作对一个路标的连续跟踪，尽管大多数情况下系统所用传感器要求距离直线非常近，从而导致机器人的运动范围只能限于直线附近。这些技术常用于工业

的自动化生产，通常将采用这些技术的车辆称为AGV(自动导引车)。一辆AGV能够在如工厂、医院和办公楼等单一场景中的预设路径上导航[78]。早期的巡线导航大多选择电缆磁带导引，但该方法的主要缺点是成本高且难以通过直线识别来重新定位导航路径。基于视觉巡线导航的早期成功方法之一是通过在采集图像的背景中提取一条白线的图像处理，并且该白线会随着车辆运动而变化[79]。在这项工作中，利用一个TV摄像头来获取环境信息。车辆的位置是由相关联的场模式变化来确定的，这时对机器人电动机提供相应的控制命令以使之沿路径规划器中预设的路径运动。在参考文献[80，81]中也提出了类似的基于预存路标(信号和直线)的车辆导航。

另一个引起广泛兴趣的研究领域是同时定位和地图建立(SLAM)，也称为同时地图构建和定位(CML)，在此移动机器人能够建立环境地图并同时根据该地图来推导自身位置。初始时，地图和车辆位置均未知，已知车辆的动力学模型，并在一个存在大量人工路标或自然路标的未知环境中运动。根据对路标的观测，同时估计机器人和路标的位置。SLAM问题中涉及对观测模型和运动模型的合理表征[82]。大多数SLAM方法都是针对完全结构化的静态环境下的室内导航[83-87]，并提供移动机器人位置和路标位置的相关度量信息。另外，还有少数一些研究工作主要是针对动态场景和室外环境的[88,89]。在早期采用声纳和视觉传感器对较大实验区域构建地图的算法中，尤为值得注意的是对特殊(如长且直的墙壁)情况的信息存储问题。随着移动机器人的运动，由于车轮打滑、地面不平整以及时钟信息校准不佳而产生的里程计信息误差会导致里程计提供的位置信息越来越偏离其实际位置。在这些位置上检测到的特征将内置在与机器人位置相关的地图中，因此，特征位置也会逐渐偏离其实际位置。现已提出一些修正上述运动偏离的算法，如参考文献[90，91]中提出保存每个特征和机器人位置之间的相关性，在参考文献[90]中提出采用扩展卡尔曼滤波(EKF)方法来实现基于视觉的SLAM。尽管EKF方法已普遍应用于解决这些问题，但都是基于传感器和处理过程的不确定性可建模为高斯分布的基本假设。然而，实际系统与这些假设条件存在显著不同。EKF实现和KF实现的一个主要缺点是对于运行时间较长的任务，路标数会不断增大，最终导致计算资源不足以实时更新地图。该问题的产生是由于每个路标都与其他路标相关联。压缩扩展卡尔曼滤波器(CEKF)算法[87]可在不影响结果准确度的条件下显著减少计算需求。CEKF算法存储和保持一个局部区域所采集的所有信息，以及与该区域中路标数二次方成正比的成本，然后以类似于完全SLAM的成本将这些信息转移到全局地图中的其余部分，但只需一次迭代运行。为解决上述问题，针对基于视觉的SLAM问题，最近提出了一些直接应用粒子滤波的方法[92,93]。然而，粒子滤波本质上是一个缓慢的处理过程，因此对于实时导航会存在很大问题。

1.5 障碍物检测和避障

避障是大多数移动机器人导航机制中一个十分重要的步骤。障碍物检测是区分地面(也称为地平面)和地面上物体的过程，即在摄像头采集图像中从障碍物像素中分离出地面像素。防撞是一个使机器人漫游且避免与障碍物碰撞的转向行为。由于本书主要是针对基于视觉的导航，因此在此仅限于讨论基于计算机视觉转向决策的研究工作。之前介绍的许多研究工作在其他传统传感器的支持下已实现避障功能，另外还有一些利用单目图像特征、立体视觉检测、光流法和基于视觉的势场法等方法来实现避障。在参考文献[94]中提出了一种基于单目摄像头图像的移动机器人自主障碍物检测方法。该系统主要包括三个视觉模块来检测障碍物。这三个视觉处理模块分别是基于亮度梯度、RGB 和 HSV，并产生一个称为障碍物边界的基于图像的粗略表示。这三个模块的输出整合成一个障碍物边界，并利用该信息来产生转弯命令。利用三个模块的目的是，在任何情况下，都有两个模块能够适合于检测边界。这种方法已经在 JPL(发射推进实验室)的两种火星模拟环境中进行了测试。然而，该系统的缺点是当障碍物超出摄像头视场范围时就会失效。

另一种避障策略是针对结构化环境的基于外观的方法[95]。该系统基于三个假设：在外观上，障碍物与地面不同；地面相对平坦；所有障碍物都与地面接触。系统采用 320×260 的彩色像素图像。处理过程主要包括四个步骤：

1）利用一个 5×5 的高斯掩膜来滤波彩色图像；

2）将滤波后的彩色图像转换到 HIS 彩色空间；

3）对梯形区域中的像素进行色度和亮度直方图化；

4）对滤波图像中的所有像素都进行色度和亮度直方图对比。如果像素色度和亮度的直方图值小于阈值，则将其看作障碍物。

该系统有三种操作模式：常规、自适应和辅助。每种模式都特别适用于一种特殊情况。在自适应模式中，系统可应对光照变化；在辅助模式中，机器人可进行远程操作。

在参考文献[96]中提出了一种利用可视化声纳进行障碍物检测和避障的方法。在该方法中，在机器人上安装一个摄像头，根据颜色可将摄像头图像中的每个像素分为地面像素、其他已知物体或未知物体。相对于机器人，以线性间隔 5°对该图像进行扫描。如果在对应于同一类颜色的一次扫描中存在一组连续像素，则可将其确定为一个物体。当未知颜色类同时出现时，则检测为未知障碍物。通过物体和地面最近相交像素点的颜色分类值之差来计算机器人与图像之间的距离，利用机器人与物体之间的距离和摄像头的有限视场来创建一个局部地图。每当在摄像头的视场中出现一个新的物体，则更新视觉信息。该视觉

算法已应用于AIBO机器人。在参考文献[97]中提出了另一种摄像头和声纳传感器相结合的障碍物检测和避障方法。在该研究工作中，利用视觉处理中的Canny边缘检测法来检测障碍物，并采用极限环和接近图导航方法来实现避障。

在基于立体视觉的障碍物检测方法中，基本思想都是同时采集环境的两幅图像。障碍物的位置可通过逆投影变换来确定[98]。在逆投影变换中，两个图像中的像素都投影到地平面，表示地面上的所有点。通过两幅图像的不同来计算障碍物的位置，这是因为这些不同表明存在障碍物。这种方法的缺点是计算量大。在参考文献[99]中也提出了室外环境下的障碍物检测和避障方法，这是通过计算一对标定的立体摄像头所采集两幅图像的视差。在该研究工作中，系统假设物体较高且表面平坦，在亮度图像中与背景区别明显。高于地面的每个点都被配置为一个可能物体，并投影在地平面中一个称为瞬时障碍地图(IOM)的局部占据栅格。根据在该瞬时障碍地图中计算的障碍物位置来产生控制机器人转向的命令。

1.6 本章小结

本章介绍了基于视觉的自主移动机器人导航的基本概念。深入讨论了几类不同的视觉导航方法，并对现今该领域内世界范围的研究工作进行了总结。

参考文献

[1] Chen, Z., Birchfield, S.T.: Qualitative Vision-Based Mobile Robot Navigation. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), Orlando, Florida (May 2006)

[2] Benet, G., Blanes, F., Simo, J.E., Perez, P.: Using infrared sensors for distance measurement in mobile robots. Robotics and Autonomous Systems 40, 255–266 (2002)

[3] Flynn, A.M.: Combining sonar and infrared sensors for mobile robot navigation. The International Journal of Robotics Research 7(6), 5–14 (1988)

[4] Saeedi, P., Lawrence, P.D., Lowe, D.G., Jacobsen, P., Kusalovic, D., Ardron, K., Sorensen, P.H.: An autonomous excavator with vision-based track-slippage. IEEE Transaction on Control Systems Technology 13(1), 67–84 (2005)

[5] Bertozzi, M., Broggi, A., Fascioli, A.: Vision-based intelligent vehicles: state of the art and perspectives. Robotics and Autonomous Systems 32, 1–16 (2000)

[6] DeSouza, G.N., Kak, A.C.: Vision for mobile robot navigation: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(2), 237–267 (2002)

[7] Shin, D.H., Singh, S.: Path generation for robot vehicles using composite clothoid segments. The Robotics Institute, Internal Report CMU-RI-TR-90-31. Carnegie-Mellon University (1990)

[8] Lebegue, X., Aggarwal, J.K.: Generation of architectural CAD models using a mobile robot. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 711–717 (1994)

[9] Lebegue, X., Aggarwal, J.K.: Significant line segments for an indoor mobile robot. IEEE Transactions on Robotics and Automation 9(6), 801–815 (1993)

[10] Egido, V., Barber, R., Boada, M.J.L., Salichs, M.A.: Self-generation by a mobile robot of topological maps of corridors. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), Washington, pp. 2662–2667 (May 2002)

[11] Borenstein, J., Everett, H.R., Feng, L. (eds.): Navigating Mobile Robots: Systems and Techniques. A. K. Peters, Wellesley (1996)

[12] Atiya, S., Hanger, G.D.: Real-time vision based robot localization. IEEE Transactions on Robotics and Automation 9(6), 785–800 (1993)

[13] Kosaka, A., Kak, A.C.: Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. Computer Vision, Graphics, and Image Processing – Image Understanding 56(3), 271–329 (1992)

[14] Meng, M., Kak, A.C.: Mobile robot navigation using neural networks and nonmetrical environment models. IEEE Control Systems, 30–39 (October 1993)

[15] Pan, J., Pack, D.J., Kosaka, A., Kak, A.C.: FUZZY-NAV: A vision-based robot navigation architecture using fuzzy inference for uncertainty. In: Proc. IEEE World Congress Neural Networks, vol. 2, pp. 602–607 (July 1995)

[16] Yamauchi, B., Beer, R.: Spatial learning for navigation in dynamic environments. IEEE Transactions on Systems, Man, and Cybernetics: Part B 26(3), 496–505 (1996)

[17] Zimmer, U.R.: Robust world-modeling and navigation in real world. In: Proc. Third International Conference Fuzzy Logic, Neural Nets, and Soft Computing, vol. 13(2-4), pp. 247–260 (October 1996)

[18] Borenstein, J., Koren, Y.: The vector-field histogram-fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation 7(3), 278–288 (1991)

[19] Elfes, A.: Sonar-based real-world mapping and navigation. IEEE Journal of Robotics and Automation 3(6), 249–265 (1987)

[20] Yagi, Y., Kawato, S., Tsuji, S.: Real-time ominidirectional image sensor (COPIS) for vision guided navigation. IEEE Transactions on Robotics and Automation 10(1), 11–22 (1994)

[21] Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence 99(1), 21–71 (1998)

[22] Martin, M.C.: Evolving visual sonar: Depth from monocular images. Pattern Recognition Letters 27(11), 1174–1180 (2006)

[23] Gartshore, R., Palmer, P.: Exploration of an unknown 2D environment using a view improvement strategy. Towards Autonomous Robotic Systems, 57–64 (2005)

[24] Santos-victor, J., Sandini, G., Curotto, F., Garibaldi, S.: Divergent stereo for robot navigation: learning from bees. In: Proc. IEEE CS Conference Computer Vision and Pattern Recognition (1993)

[25] Ohno, T., Ohya, A., Yuta, S.: Autonomous navigation for mobile robots referring pre-recorded image sequence. In: Proc. IEEE International Conference on Intelligent Robots and Systems, vol. 2, pp. 672–679 (November 1996)

[26] Jones, A.D., Andersen, C., Crowley, J.L.: Appearance based processes for visual navigation. In: Proc. IEEE International Conference on Intelligent Robots and Systems, pp. 551–557 (September 1997)
[27] Talukder, A., Goldberg, S., Matties, L., Ansar, A.: Real-time detection of moving objects in a dynamic scene from moving robotic vehicles. In: Proc. IEEE International Conference on Intelligent Robots and Systems (IROS), Las Vegas, Nevada, pp. 1308–1313 (October 2003)
[28] Talukder, A., Matties, L.: Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. In: Proc. IEEE International Conference on Intelligent Robots and Systems (IROS), Sendai, pp. 3718–3725 (October 2004)
[29] Braillon, C., Usher, K., Pradalier, C., Crowley, J.L., Laugier, C.: Fusion of stereo and optical flow data using occupancy grid. In: Proc. IEEE International Conference on Intelligent Robots and Systems (IROS), Beijing, pp. 2302–2307 (October 2006)
[30] Matsumoto, Y., Ikeda, K., Inaba, M., Inoue, H.: Visual navigation using omnidirectional view sequence. In: Proc. IEEE International Conference on Intelligent Robots and Systems (IROS), Kyongju, Korea, pp. 317–322 (October 1999)
[31] Thorpe, C., Herbert, M.H., Kanade, T., Shafer, S.A.: Vision and Navigation for the Carnegie-Mellon Navlab. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(3), 362–372 (1988)
[32] Thorpe, C., Kanade, T., Shafer, S.A.: Vision and Navigation for the Carnegie-Mellon Navlab. In: Proc. Image Understand Workshop, pp. 143–152 (1987)
[33] Broggi, A., Berte, S.: Vision-based road detection in automotive systems: A real-time expectation-driven approach. Journal of Artificial Intelligence Research 3(6), 325–348 (1995)
[34] Ghurchian, R., Takahashi, T., Wang, Z.D., Nakano, E.: On robot self navigation in outdoor environments by color image processing. In: Proc. International Conference on Control, Automation, Robotics and Vision, pp. 625–630 (2002)
[35] Jung, C.R., Kelber, C.R.: Lane following and lane departure using a linear-parabolic model. Image and Vision Computing 23(13), 1192–1202 (2005)
[36] Schneiderman, H., Nashman, M.: A discriminating feature tracker for vision-based autonomous driving. IEEE Transactions on Robotics and Automation 10(6), 769–775 (1994)
[37] Mejias, L.O., Saripalli, S., Sukhatme, G.S., Cervera, P.C.: Detection and tracking of external features in an urban environment using an autonomous helicopter. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), Barcelona, pp. 3972–3977 (April 2005)
[38] Saeedi, P., Lawrence, P.D., Lowe, D.G.: Vision-based 3-D trajectory tracking for unknown environments. IEEE Transaction on Robotics 22(1), 119–136 (2006)
[39] Moravec, H.P.: The stanford cart and the CMU rover. Proc. IEEE 71(7), 872–884 (1983)
[40] Thorpe, C.: FIDO: Vision and navigation for a mobile robot. PhD dissertation, Department of computer science, Carnegie Mellon University (December 1984)
[41] Horswill, I.: Visual collision avoidance by segmentation. In: Proc. IEEE International Conference on Intelligent Robots and Systems, Germany, pp. 902–909 (September 1994)

[42] Horswill, I.: Specialzation of Perceptual Processes. PhD thesis, Massachusetts Institute of Technology (1995)
[43] Ohya, A., Kosaka, A., Kak, A.: Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. IEEE Transactions on Robotics and Automation 14(6), 969–978 (1998)
[44] Aider, O.A., Hoppenot, P., Colle, E.: A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences. Robotics and Autonomous Systems 52, 229–246 (2005)
[45] Gartshore, R., Aguado, A., Galambos, C.: Incremental map building using occupancy grid for an autonomous monocular robot. In: Proc. Seventh International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, pp. 613–618 (December 2002)
[46] Murillo, A.C., Kosecka, J., Guerrero, J.J., Sagues, C.: Visual door detection integrating appearance and shape cues. Robotics and Autonomous Systems 56, 512–521 (2008)
[47] Saitoh, T., Tada, N., Konishi, R.: Indoor mobile robot navigation by center following based on monocular vision. In: Computer Vision, pp. 352–366. In-teh Publishers
[48] Birchfield, S.: KLT: An implementation of the Kanade- Lucas-Tomasi feature tracker, http://www.ces.clemson.edu/~stb/klt/
[49] Kidono, K., Miura, J., Shirai, Y.: Autonomous visual navigation of a mobile robot using a human guided experience. Robotics and Autonomous Systems 40(23), 124–132 (2002)
[50] Murray, D., Little, J.J.: Using real-time stereo vision for mobile robot navigation. Autonomous Robots 8, 161–171 (2000)
[51] Davison, A.J.: Mobile robot navigation using active vision. PhD thesis (1998)
[52] Ayache, N., Faugeras, O.D.: Maintaining representations of the environment of a mobile robot. IEEE Transactions on Robotics and Automation 5(6), 804–819 (1989)
[53] Olson, C.F., Matthies, L.H., Schoppers, M., Maimone, M.W.: Rover navigation using stereo ego-motion. Robotics and Autonomous Systems 43(4), 215–229 (2003)
[54] Konolige, K., Agrawal, M., Bolles, R.C., Cowan, C., Fischler, M., Gerkey, B.: Outdoor Mapping and Navigation using Stereo Vision. In: Proc. International Symposium on Experimental Robotics (ISER), Brazil, pp. 1–12 (July 2006)
[55] Shi, J., Tomasi, C.: Good Features to Track. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994), Seattle, pp. 593–600 (June 1994)
[56] Nishimoto, T., Yamaguchi, J.: Three dimensional measurements using fisheye stereo vision. In: Proc. SICE Annual Conference, Japan, pp. 2008–2012 (September 2007)
[57] Yamaguti, N., Oe, S., Terada, K.: A Method of distance measurement by using monocular camera. In: Proc. SICE Annual Conference, Japan, pp. 1255–1260 (July 1997)
[58] Chou, T.N., Wykes, C.: An integrated ultrasonic system for detection, recognition and measurement. Measurement 26, 179–190 (1999)

[59] Conradt, J., Simon, P., Pescatore, M., Verschure, P.F.M.J.: Saliency Maps Operating on Stereo Images Detect Landmarks and Their Distance. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 795–800. Springer, Heidelberg (2002)
[60] Wooden, D.: A guide to vision-based map-building. IEEE Robotics and Automation Magazine, 94–98 (June 2006)
[61] Goldberg, S.B., Maimone, M.W., Matthies, L.: Stereo vision and rover navigation software for planetary exploration. In: Proc. IEEE Aerospace Conference Proceedings, USA, vol. 5, pp. 5025–5036 (March 2002)
[62] Fialaa, M., Basub, A.: Robot navigation using panoramic tracking. Pattern Recognition 37, 2195–2215 (2004)
[63] Gasper, J., Santos- Victor, J.: Vision-based navigation and environmental representations with an omnidirectional camera. IEEE Transactions on Robotics and Automation 16(6), 890–898 (2000)
[64] Winters, N., Santos-victor, J.: Ominidirectional visual navigation. In: Proc. IEEE International Symposium on Intelligent Robotic Systems (SIRS), pp. 109–118 (1999)
[65] Gasper, J., Winters, N., Santos-victor, N.: Vision-based navigation and environmental representation with an ominidirectional camera. IEEE Transtations on Robotics and Automation 16(6), 890–898 (2000)
[66] Srinivasan, M.V.: An image-interpolation technique for the computation of optic flow and Egomotion. Biological Cybernetics 71(5), 401–415 (1994)
[67] Srinivasan, M.V., Zhang, S.: Visual navigation in flying insects. International Review of Neurobiology 44, 67–92 (2000)
[68] Coombs, D., Roberts, K.: Centering behaviour using peripheral vision. In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, USA, pp. 440–445 (June 1993)
[69] Sandini, G., Santos-Victor, J., Curotto, F., Garibaldi, S.: Robotic bees. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, Yokohama, Japan, vol. 1, pp. 629–635 (1993)
[70] Santos-Victor, J., Sandini, G.: Divergent stereo in autonomous navigation: From bees to robots. International Journal of Computer Vision 14(2), 159–177 (1995)
[71] Lourakis, M.I.A., Orphanoudakis, S.C.: Visual Detection of Obstacles Assuming a Locally Planar Ground. In: Chin, R., Pong, T.-C. (eds.) ACCV 1998. LNCS, vol. 1352, pp. 527–534. Springer, Heidelberg (1997)
[72] Camus, T.: Real-time quantized optical flow. Real-Time Imaging 3(2), 71–86 (1997)
[73] Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. DARPA Image Understanding Workshop, pp. 121–130 (1984)
[74] Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 13, 185–203 (1981)
[75] Nagel, H.: On the estimation of optical flow: relations between different approaches and some new results. Artificial Intelligence 33(3), 299–324 (1987)
[76] van der Zwaan, S., Santos-Victor, J.: An insect inspired visual sensor for the autonomous navigation of a mobile robot. In: Proc. Seventh International Sysposium on Intelligent Robotic Systems, Portugal (July 1999)

[77] Netter, T., Franceschini, N.: A robotic aircraft that follows terrain using a neuromorphic eye. In: Proc. IEEE International Conference on Intelligent Robots and Systems (IROS), Switzerland, vol. 1, pp. 129–134 (Septemper 2002)
[78] Zhang, H., Yuan, K., Mei, S., Zhou, Q.: Visual navigation of automated guided vehicle based on path recognition. In: Proc. Third International Conference on Machine Learning and Cybernectics, Shanghai, pp. 26–29 (August 2004)
[79] Ishikawa, S., Kuwamoto, H., Ozawa, S.: Visual navigation of an autonomous vehicle using white line recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(5), 743–749 (1988)
[80] Beccari, G., Caselli, S., Zanichelli, F., Calafiore, A.: Vision-based line tracking and navigation in structured environments. In: Proc. IEEE International Symposium on Computational Intelligent in Robotics and Automation, USA, pp. 406–411 (July 1997)
[81] Ismail, A.H., Ramli, H.R., Ahmad, M.H., Marhaban, M.H.: Vision-based system for line following mobile robot. In: Proc. IEEE Symposium on Industrial Electronics and Applications (ISIEA), Malaysia, pp. 642–645 (October 2009)
[82] Durrant-White, H., Bailey, T.: Simultaneous localization and mapping. IEEE Robotics and Automation Magazine 13(2), 99–108 (2006)
[83] Zunino, G., Christensen, H.I.: Simultaneous localization and mapping in domestic environments. Multisensor Fusion and Integration for Intelligent Systems, 67–72 (2001)
[84] Bosse, M., Newman, P., Leonard, J., Teller, S.: Slam in large-scale cyclic environments using the atlas framework. International Journal of Robotics Research 23(12), 1113–1139 (2004)
[85] Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., Csorba, M.: A solution to the simultaneous localization and map building (slam) problem. IEEE Transactions on Robotics and Automation 17(3), 229–241 (2001)
[86] Estrada, C., Neira, J., Tardos, J.D.: Hierarchical SLAM: Real-time accurate mapping of large environments. IEEE Transactions on Robotics 21(4), 588–596 (2005)
[87] Guivant, J.E., Nebot, E.M.: Optimization of the simultaneous localization and map-building algorithm for real-time implementation. IEEE Transactions on Robotics and Automation 17(3) (June 2001)
[88] Andrade-Cetto, J., Sanfeliu, A.: Concurrent map building and localization on indoor dynamic environment. International Journal of Pattern Recognition and Artificial Intelligence 16(3), 361–374 (2002)
[89] Liu, Y., Thrun, S.: Results for outdoor-SLAM using sparse extended information filters. In: Proc. IEEE Conference on Robotics and Automation (ICRA), Taipei, pp. 1227–1233 (September 2003)
[90] Davison, A.J., Murray, D.: Simultaneous localization and map-building using active vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 865–880 (2002)
[91] Newman, P., Bosse, M., Leonard, J.: Autonomous feature-based exploration. In: Proc. International Conference on Robotics and Automation (ICRA), Taipei, vol. 1, pp. 1234–1240 (September 2003)
[92] Sim, R., Elinas, P., Griffin, M., Little, J.J.: Vision based SLAM using the Rao-Blackwellized particle filter. In: Proc. IJCAI Workshop Reasoning with Uncertainty in Robotics, Edinburgh, Scotland (July 2005)

[93] Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, pp. 1151–1156 (August 2003)

[94] Lorigo, L.M., Brooks, A., Grimson, W.E.L.: Visually-guided obstacle avoidance in unstructured environments. In: Proc. IEEE Conference on Intelligent Robots and Systems, France (1997)

[95] Ulrich, I., Nourbakhsh, I.: Appearance-based obstacle detection with monocular colour vision. In: Proc. AAAI Conference on Artificial Intelligence, USA (July 2000)

[96] Lenser, S., Veloso, M.: Visual Sonar: Fast obstacle avoidance using monocular vision. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, pp. 886–891 (2003)

[97] Kim, P.G., Park, C.G., Jong, Y.H., Yun, J.H., Mo, E.J., Kim, C.S., Jie, M.S., Hwang, S.C., Lee, K.W.: Obstacle Avoidance of a Mobile Robot Using Vision System and Ultrasonic Sensor. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. LNCS, vol. 4681, pp. 545–553. Springer, Heidelberg (2007)

[98] Bertozzi, M., Broggi, A., Fascioli, A.: Real-time obstacle detection using stereo vision. In: Proc. VIII European Signal Processing Conference, Italy, pp. 1463–1466 (September 1996)

[99] Badal, S., Ravela, S., Draper, B., Hanson, A.: A practical obstacle detection and avoidance system. In: Proc. 2nd IEEE Workshop on Application of Computer Vision, pp. 97–104 (1994)

[100] Nirmal Singh, N.: Vision Based Autonomous Navigation of Mobile Robots. Ph.D. Thesis, Jadavpur University, Kolkata, India (2010)

第 2 章　移动机器人的外设接口㊀

摘要：本章介绍了如何在现有移动机器人上实现与外设的实际接口。由于经费限制，很多情况下都无法购置一个具有所有外设和传感器系统的完整机器人，因此希望该系统在这种研究条件下有很大帮助。本章讨论了如何利用串口通信以中断驱动模式实现与 KOALA 机器人的外设接口。

2.1　简介

如今，许多实际移动机器人都配置有一些能够增强机器人系统先进性和灵活性并有助于整体性能和智能化程度的传感器和其他辅助设备。在许多情况下，要提高自动化程度需要与一些实际驱动的附加外设接口。机器人随时对这些传感器和辅助设备发出控制指令，而这些外围设备应在满足系统实时性需求的条件下响应机器人请求。因此，利用机器人核心部件以及其他附件成功开发一个集成系统需要复杂的中断驱动软件程序。目前，已可为世界范围从事机器人研究的人员提供多种不同自动化程度的机器人平台，大部分这些机器人平台都配置有多个传感器和辅助设备以及能够实现内/外部实时通信的必要软件。然而，如果用户想要在这些机器人平台上增加一些不是由同一生产厂商支持/制造的传感器和其他辅助设备作为外设时，就不很方便。通常情况下没有移动机器人外设接口装置的技术手册，这是因为机器人生产厂商并没有提供这些技术资料，即便这些附加传感器是与机器人套件一起购买的。

近年来，基于 PIC 微控制器的系统在一些研究领域得到了广泛的实际应用，如变容二极管的人工神经网络模型的硬件开发[1]、基于 Petri 网的过程和状态监测分布式系统[2]、原子碰撞实验中常用的双射束调制系统的开发[3]、递归神经网络模型的硬件实现[4]、模糊控制的同步电动机的无功功率控制[5]等。PIC 微控制器的架构是基于一种改进的哈佛 RISC 指令集[6,1]。由于可提供出色的低成本解决方案和先进性能，该技术越来越得到广泛应用。鉴于数据和指令的传输都

㊀ 本章基于 N. Nirmal Singh、Amitava Chatterjee 和 Anjan Rakshit 2010 年发表于 International Journal of Electronics 期刊第 97 期，139 ~ 161 页的论文“A PIC Microcontroller system for Real-life Interfacing of External Peripherals with a Mobile Robot”。经出版商许可后重印（Taylor & Francis Ltd，http: //www. tandf. co. uk/journals）。

是在独立总线上实现的，因此能够提供满意的性能。这些处理器还能够提高软件代码效率，并在取下一条指令的同时执行当前指令[1]。本章将详细讨论如何开发一个基于 PIC 微控制器的系统，使之能够实现目前市场上已有的常见移动机器人与外设接口[10,11]。在此所采用的移动机器人是由瑞士 K-team S. A 公司生产的 KOALA 机器人。印度加尔各答贾达普(Jadavpur)大学电子工程学院电子测量和电子设备实验室购置的 KOALA 机器人仅配置有增量编码器、声纳传感器和红外传感器。但是为了能够利用视觉感知对 KOALA 移动机器人进行导航，需要对机器人开发并集成一个视觉感知系统的接口装置。因此，最初的研究工作是直接在 KOALA 机器人上分别增加一个立体视觉装置(由两个摄像头组成)和一个单目视觉装置(只由一个摄像头组成)，这样就使得机器人在导航时可灵活选择双摄像头系统或单摄像头系统。本章主要讨论在 KOALA 机器人外部集成的视觉系统中增加一个四自由度(DOF)的立体视觉装置，开发一个用于机器人系统云台控制、左右摄像头聚焦控制的 PIC 微控制器系统。同时还详细介绍了以中断驱动模式开发的应用软件，有助于其他用户开发类似的集成系统。这样可使得研究人员/开发人员对机器人保持完全灵活性，同时开发成本也会显著减少。事实上，这项研究工作的主要动机是由于预算不足以购买集成了视觉系统的 KOALA 机器人完整套件，因此只能在实验室自主实现 KOALA 机器人与视觉系统的集成。在此也真心希望这项工作能够鼓励机器人领域的其他研究人员能够自主开发这种能够使得独立外设与其他机器人套件接口的中断驱动系统。这样将有助于开发高复杂度的低成本机器人平台，尽管现有系统仅具有四个外设(即四自由度的四个 RC 伺服电动机) 接口，但从逻辑上可扩展更多的外设。

2.2 用于视觉系统与现有机器人接口的 PIC 微控制器系统

本节所开发的系统采用一个 PIC 16F876A 微控制器来实现 KOALA 机器人与一个视觉系统的实时接口，目的是为了实现四个 RC 伺服电动机[7]与 KOALA 移动机器人的实际接口，由此可对集成于 KOALA 机器人中的视觉系统增加四自由度。图 2.1 给出本实验室开发的集成了视觉系统的 KOALA 机器人完整系统。其中，包括内置有红外、增量编码器等传感器的基本 KOALA 机器人。为提高机器人系统的能力，另外还集成了两个声纳传感器和两个摄像头来实现立体视觉功能。然而，如前所述，在 KOALA 机器人中开发的视觉导航系统还配置了独立的单目视觉功能。在利用立体视觉开发的系统和利用单目视觉开发的系统之间只有一个显著差别。单目视觉情况下，在主动式云台系统的中心只安装一个摄像头，且系统只利用两个自由度来实现云台控制。所开发的集成视觉系统应能够实现对于立体视觉四自由度和对于单目视觉两自由度的灵活控制。因此，为提

高视觉系统的高度灵活性，一个云台系统集成了四个伺服电动机。图 2.2 给出用于驱动四个伺服电动机的 PIC 微控制器的原理示意图[7]。

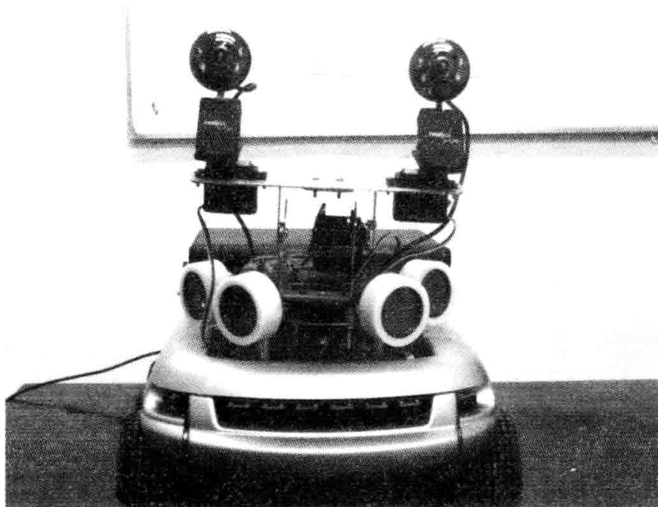


图 2.1　KOALA 机器人的完整视觉系统

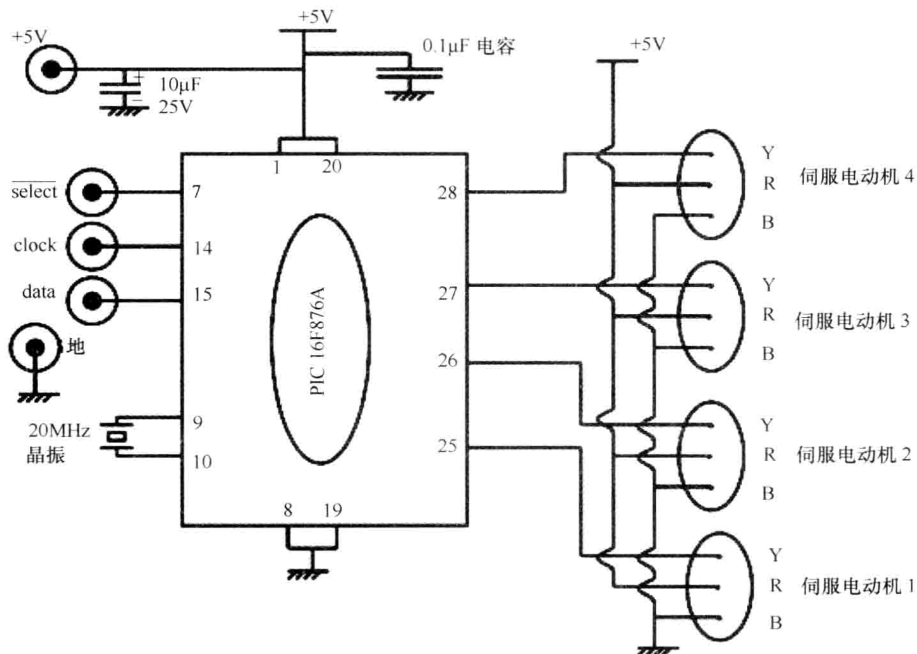


图 2.2　用于 KOALA 机器人与四个伺服电动机接口的 PIC 16F876A 控制板的原理示意图（Y：黄色；R：红色；B：黑色）

利用四个 RC 伺服电动机来实现完整视觉系统的云台控制，以及左右摄像头的各自聚焦控制。PIC 16F876A 微控制器接收来自于 Motorola 68331 处理器中三个输入引脚[$\overline{\text{select}}$(片选信号)、clock(时钟信号)和 data(数据信号)]的信号。68331 处理器工作在 SPI 主模式，而 PIC 16F876A 微控制器工作在 SPI 从模式。图 2.3 给出了所用 PIC 微控制器板的实物图片。该控制器板安装在与云台系统支架相垂直的位置上，使得整个集成系统的结构足够坚固。

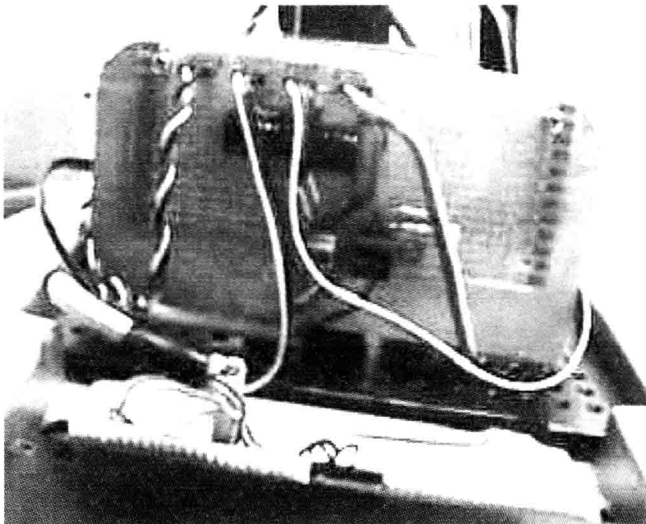


图 2.3 所用 PIC 微控制器板的实物图片

如前所述，该系统采用了一个 28 引脚双列直插式封装(PDIP)的 PIC 16F876A 微控制器。PIC 16F876A 微控制器的主要特点是[6]包含 8KB 的 flash 程序存储器、368B 的数据存储器和 256B 的 EEPROM 数据存储器。工作频率为 0Hz~20MHz，并提供 14 种中断。PIC 16F876A 微控制器具有三个 I/O 接口(A、B 和 C)、三个定时器、两个模拟比较器和五个 10bit A-D 转换输入通道。通过 MSSP 和 USART 实现串口通信。图 2.4 给出了 28 引脚双列直插式封装 PIC 16F876A 微控制器的引脚图。

算法 2.1 描述了实现 KOALA 机器人与外设实际接口操作(本例中为视觉系统的 RC 伺服电动机)的 PIC 微控制器主程序算法。其中，介绍了如何对 PORTB 的数据方向寄存器、A-D 转换控制寄存器、MSSP 控制寄存器、INTCON 寄存器、PIE1 寄存器和 PIR1 寄存器进行编程以及具体的编程顺序。不断对系统初始化，使得在每个 RC 伺服电动机的中间位置初始化并等待一个合适的输入驱动指令。然后，系统允许同步串口中断(SPI)，而对于 PIC 微控制器，在从模式下设置 SPI。接下来，根据中断服务标志位，由具体指令驱动相应的 RC 伺服电动机。然后，复位中断服务标志位，以使下次新的中断请求发生时可进行设置。KOA-

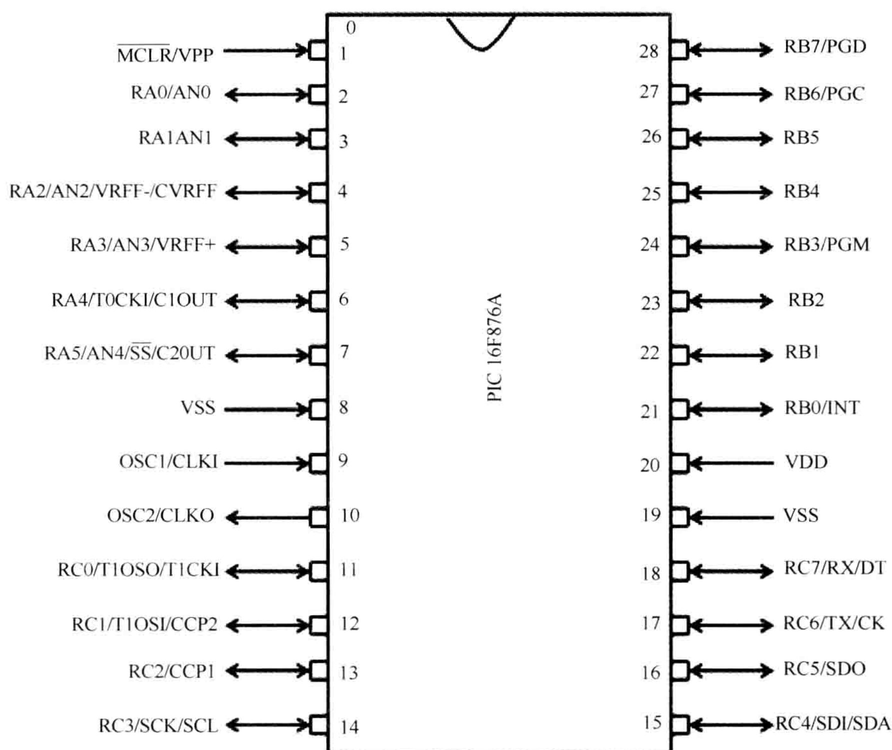


图 2.4　28 引脚双列直插式封装 PIC 16F876A 微控制器的引脚图[6]

LA 机器人的主要技术特点[8]是包括一个工作频率为 22MHz 的 Motorola 68331 处理器；机器人的 RAM 容量为 1MB，且闪存容量也是 1MB，但没有 ROM；为实现运动，KOALA 机器人配置了具有增量编码器的直流电动机；与车轮相连的直流电动机通过一个 58.5∶1 的减速齿轮来实现车轮运动 KOALA 机器人的主处理器具有直接控制电动机供电电源的装置；该处理器可读取增量编码器的脉冲信号；RS232 串口通信总是设置为包括一个起始位、两个停止位和无奇偶校验的 8bit 模式；波特率变化范围为 9600 ~ 115200；机器人具有十二个数字输入接口、四个 COMS/TTL 数字输出接口、八个电源数字输出接口和六个模拟输入接口；机器人本体配置有十六个红外接近传感器。这些红外传感器中包括一个红外 LED 和一个接收器，由德州仪器公司制造（TSL252 型），可用于环境光测量和反射光测量，其输出值是一个由 10bit A-D 转换器转换的模拟量。因此，在 KOALA 机器人的基本架构上，集成了一个由双目摄像头、云台系统、四个 RC 伺服电动机和 PIC 微控制器板构成的视觉系统，从而构成了一个改进型机器人。

Motorola MC68331 处理器是一个 32bit 微控制器，具有与外设之间的高速数据处理能力。该微控制器包括一个 32bit CPU、一个系统集成模块、一个通用计

时器(GPT)和一个队列串行模块(QSM)。MC68331控制单元的一个主要优点是低功耗。CPU基于工业级的MC68000处理器，综合了MC68010和MC68020处理器的许多特征，并增加了高性能控制器应用的特殊功能。通过采用11通道的GPT可实现中等规模的CPU控制。这些GPT引脚还可配置为通用的I/O接口。QSM由两个串行接口组成：队列串行外设接口(QSPI)；串行通信接口(SCI)。QSPI可提供简单的外设扩展或处理器间通信。

算法2.1 用于外设实际接口的PIC微控制器主程序算法

---

1. 初始化PORT B。

2. 设置对应于PORT B的数据方向寄存器，使得PORT B的4-7引脚配置为数字输出接口。

3. 设置A-D转换控制寄存器的内容为1，使得A-D转换接口的所有引脚都配置为数字I/O接口。

4. 设置角位置指令，使得每个RC伺服电动机都位于中间位置，即0°位置。

5. 设置主同步串口(MSSP)控制寄存器为1(在SPI模式下)，使得同步串口使能，并且SCK、SDO、SDI和SS配置为串口引脚；SPI设置为从模式，且SS引脚控制使能。

6. 设置MSSP状态寄存器（在SPI模式下)，使得重置SPI时钟信号选择位。

7. 设置外设中断使能寄存器为1，使得同步串口(SSP)中断使能。

8. 设置INTCON寄存器，使能所有未屏蔽外设中断和全局未屏蔽中断。

9. 重置PIR1寄存器的SSP中断标志位来表明初始时不会发生SSP中断。

10. IF 设置中断服务标志位

    THEN

        检查RC伺服电动机的ID，以确定产生何种中断。

        综合八位寄存器值、高字节数据和低字节数据，来产生该伺服电动机的16位角位置指令。

        重置中断服务标志位。

    ENDIF

11. 设 $j=1$。

12. FOR $j=1$ to 4，

    使得对RC伺服电动机(j)产生驱动指令的PORT B的RBx引脚为高电平。

    保持RBx一段时间内为高电平，作为其相应角位置指令的函数进行计算。

    之后将RBx设置为低电平。

    ENDFOR

13. 保持每个RB4~RB7引脚为低电平20ms。

14. 清空 CPU 的看门狗。
15. 返回至第 10 步。

算法 2.1 与同步串口中断子程序的算法 2.2 共同运行。这是在实际外设应用中所执行的一个非常复杂的过程。在此，算法 2.2 阐述了如何读取输入驱动命令来执行特定的驱动命令。角位置命令为 2B，且高字节和低字节均以串行方式传输。实际上，发送的第一个信息是驱动所需的 RC 伺服电动机 ID，接下来是串行的高字节和低字节的位置命令，用于指定 RC 伺服电动机应旋转多少度。图 2.2 给出了外部时钟频率为 20MHz 的 PIC 16F876A 微控制器，这是 PIC 16F876A 处理器的最大允许工作频率，且最高允许值用于实现该特定应用的最佳性能。PIC 处理器使用三个输入信号：select(用于触发)、clock(时钟信号)和 data(确定电动机驱动及其角位置的输入波形数据)。PIC 处理器的输出取为 PORTB 的四个引脚，分别为每个 RC 伺服电动机产生各自的角位置命令。

算法 2.2　同步串行中断服务子程序算法

---

1. IF count_flag = 0
   THEN
   读取 SSP 接收/发送缓存寄存器的值作为 RC 伺服电动机 ID。
   ENDIF
2. IF count_flag = 1
   THEN
   读取 SSP 接收/发送缓存寄存器的值作为“高字节数据”，即角位置命令的高字节。
   ENDIF
3. IF count_flag = 2
   THEN
   读取 SSP 接收/发送缓存寄存器的值作为“低字节数据”，即角位置命令的低字节。
   ENDIF
4. count_flag 加 1。
5. IF count_flag > 2
   THEN
   重置 count_flag 为 0。
   设置中断服务标志位。
   ENDIF

6. 重置 PIR1 寄存器的 SSP 中断标志位。

在算法 2.1 中，步骤 1 ~ 9 用于初始化。在开始时，对 8 位双向接口[6]的 PORTB 初始化。PORTB 对应的数据方向寄存器记为 TRISB。通过合理设置 TRISB 寄存器，可为数字输入或数字输出设置每个相应的 PORTB 引脚。在本实验中，对 PORTB 编程实现其高四位引脚，即引脚 4 ~ 7(记为 RB4 ~ RB7)配置为数字输出接口，用于产生四个伺服电动机的角位置命令。

PIC 微控制器的另一个重要模块是 A-D 转换模块。对于 PIC 16F876A 微控制器，该模块具有五个输入接口。在此，对于某一给定模拟输入信号，该模块产生一个 10 位数字作为输出。另外，该模块还具有四个寄存器：A-D 转换结果高位寄存器(ADRESH)；A-D 转换结果低位寄存器(ADRESL)；A-D 转换控制寄存器 0(ADCON0)；A-D 转换控制寄存器 1(ADCON1)。ADCON1 寄存器可用来设置接口引脚为模拟输入接口或数字 I/O 接口。图 2.5 给出对 ADCON1 寄存器的描述[6]。对于所开发的系统，对 A-D 转换接口设置控制位编程，使得这些引脚均设置为数字 I/O 接口。

在 PIC 16F876A 微控制器中，与其他外设或微控制器设备的串口通信由主同步串口(MSSP)模块控制实现。MSSP 模块是一个可与 EEPROM、A-D 转换寄存器、移位寄存器等进行通信的串行接口。MSSP 模块可工作于串行外设接口(SPI)模式或内置集成电路(I^2C)模式。在本实验中，在 SPI 模式下进行系统开发，这时以一种同步方式同时发送和接收八位数据。在 SPI 工作模式下，MSSP 模块中使用了一个状态寄存器(SSPSTAT)和两个控制寄存器(SSPCON 和 SSPCON2)。另外，还有两个其他寄存器，即串行接收/发送缓存寄存器(SSPBUF)和 MSSP 移位寄存器(SSPSR)。在这些寄存器中，SSPSR 寄存器不能直接访问，只能通过定位 SSPBUF 寄存器来访问。在本系统中，PIC 16F876A 处理器工作在从模式，而 KOALA 机器人中的 Motorola 68331 处理器工作在主模式。图 2.6 给出了 SSPSTAT 和 SSPCON 寄存器的具体描述。通常，有三个引脚用于通信：串行数据输出(SDO)；串行数据输入(SDI)；串行时钟(SCK)。当工作在 SPI 从模式时，还要使用第 4 个引脚从选择($\overline{SS}$)。在 SPI 工作模式，对 SSPCON 寄存器编程，使得 SSPEN 位为 1，使能串口，并配置 SCK、SDO、SDI 和 SS 引脚为串口引脚。同理，对 SSPM3:SSPM0 位编程，使得 PIC 处理器配置为 SPI 从模式，且将其时钟信号赋予 SCK 引脚。因此，在图 2.2 中，$\overline{\text{select}}$对应于$\overline{SS}$引脚，时钟信号在 SCK 引脚，以及数据位于 SDI 引脚。这样就能确保数据传输发生在从活动状态到空闲时钟状态的转移过程中。图 2.7 给出了在本系统中实现 Motorola 处理器与 PIC 处理器接口的 SPI 主/从连接编程。在接收数据时，由于 PIC 处理器是在从模式下编程，SSPSR 和 SSPBUF 寄存器共同创建一个双缓存接收

器。在此，SSPSR 移位寄存器用于移入或移出数据(MSB 优先)，SSPBUF 寄存器是数据字节写入或读出的缓存寄存器。

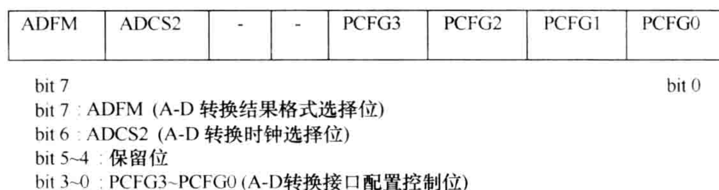


图 2.5　ADCON1 寄存器描述[6]



a)



b)

图 2.6　SSPSTAT 和 SSPCON 寄存器描述(SPI 模式)[6]

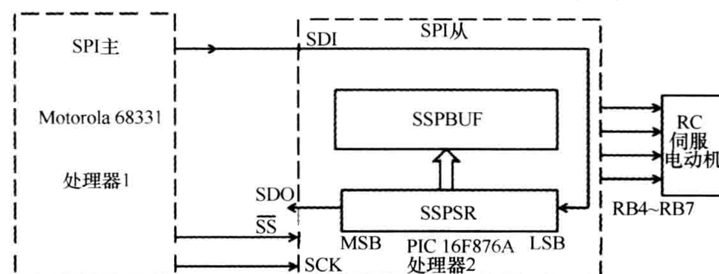


图 2.7　系统中编程实现 SPI 主/从连接

接下来，置 PIE1 寄存器的 SSPIE 位为 1，以使能同步串口(SSP)中断。PIE1 寄存器中包括单独的外设中断使能位。在此应注意到，INTCON 寄存器的 PEIE

位必须置为 1 以使能任何未屏蔽外设中断。从用户编程角度来看，INTCON 寄存器非常重要，这是由于该寄存器中包含几个 TMR0 寄存器溢出、RB 接口改变和外部 RB0/INT 引脚中断的使能位和标志位。图 2.8 分别给出 PIE1 寄存器和 INTCON 寄存器的详细描述。在程序中，INTCON 寄存器中的 PEIE 位和 GIE 位均设为 1。应注意到无论 GIE 位及其相应使能位的状态如何，在中断条件发生时，都需设置中断标志位为 1。因此，编写用户程序时，在中断使能之前，必须注意要重置合适的中断标志位。因此，在初始时重置 PIR1 寄存器的 SSPIF 位，以表明没有发生 SSP 中断条件且系统已准备好 SSP 中断使能。PIR1 寄存器是一个包含外设中断独立标志位的特殊寄存器，因此编写用户程序时，在一个特殊中断使能之前，必须重置 PIR1 寄存器中相应的标志位。图 2.9 给出了 PIR1 寄存器的详细描述。

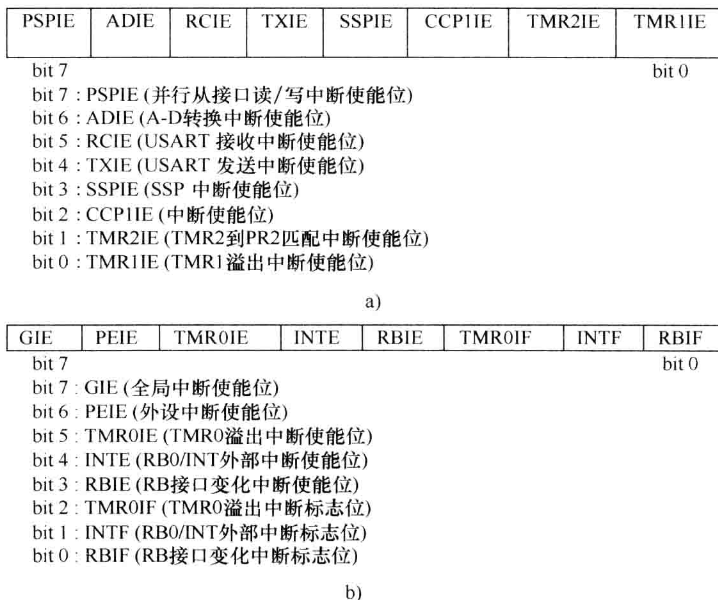


图 2.8　PIE1 和 INTCON 寄存器描述



图 2.9　PIR1 寄存器描述

一旦完成初始化，算法 2.1 所述的主程序中步骤 10 ~ 15 与算法 2.2 中的串行中断服务子程序共同执行一个无限循环。在中断子程序中，PIC 处理器在从模式下等待 Motorola 处理器写入/发送的数据。图 2.10 给出接收到的 SPI 写数据形式。波形表明一个字节数据的写入，对每个写入的字节，即 PIC 处理器接收的每个字节，不断重复执行该过程。对系统编程使得每个数据位都可在时钟信号的上升沿或下降沿锁存。在主模式下，数据传输总是由 Motorola 处理器通过发送 SCK 信号进行初始化。PIC 处理器的$\overline{SS}$引脚为低电平，则使能数据发送和接收操作。而在从模式下，若$\overline{SS}$引脚为高电平或通过清除 SSPEN 位来重置 SPI 模块。在本系统中，SPI 模块是通过强制设置$\overline{SS}$引脚为高电平来重置的。SSPBUF 寄存器保存将要写入 SSPSR 寄存器的数据，直到所接收的数据准备好。如前所述，数据是按字节一个一个地接收，对于每个接收的字节，一旦接收了 8bit 数据，则将该字节转移到 SSPBUF 寄存器。通过设置缓存区满检测位(BF)和中断标志位(SSPIE)为 1 来标记该操作。因此，这种双缓存机制能够在全部读入刚接收的数据字节之前开始接收新数据字节。由 Motorola 处理器发送，然后由 PIC 处理器在该串行中断模式下接收的数据字节序列按如下编程：

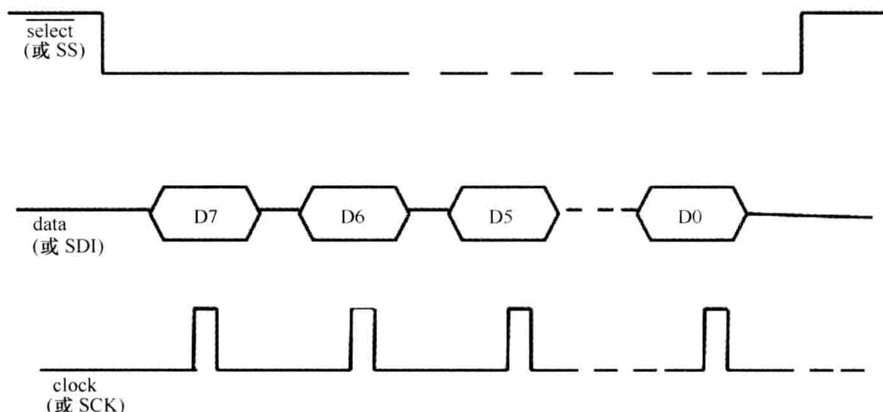


图 2.10　SPI 模式的波形(从模式)

1)发送包含 RC 伺服电动机 ID(1 ~ 4)的数据字节。

2)延迟 10ms。

3) 对特定 RC 伺服电动机 ID 发送相应角位置命令的高字节。

4)延迟 10ms。

5) 对同一 RC 伺服电动机 ID 发送相应角位置命令的低字节。

在 1)、3)和 5)的每个步骤中，SSPBUF 寄存器读入不同的临时变量。当完成 1) ~ 5)步骤时，也就实现了一次完整的信息传输周期。通过在软件中设置一个临时标志变量(中断服务标志位)来进行标记。在从中断服务子程序返回之前，用户程序还必须重置 PIR1 寄存器中的 SSP 中断标志位(SSPIF)。一旦在主程序

中设置了该中断服务标志位，则合并 RC 伺服电动机角位置命令的高字节和低字节来创建一个合适的角位置命令，同时对 RC 伺服电动机发送一个相应的驱动命令。对于每个 RC 伺服电动机，通过驱动相应的引脚信号以维持一段时间的高电平，然后驱动同一引脚信号维持一段时间的低电平来产生该数字驱动信号。1.5ms 的高电平信号对应于 0°的角位置命令，1ms 的高电平信号则对应 -90°，2ms 时对应 +90°。对于所有中间的角位置命令，发送一个成正比的高电平信号维持时间。高电平信号之后总是一个 20ms 固定时间的低电平信号。图 2.11 给出了在 PIC 16F876A 处理器的 RB4、RB5、RB6 和 RB7 引脚处产生的输出波形。

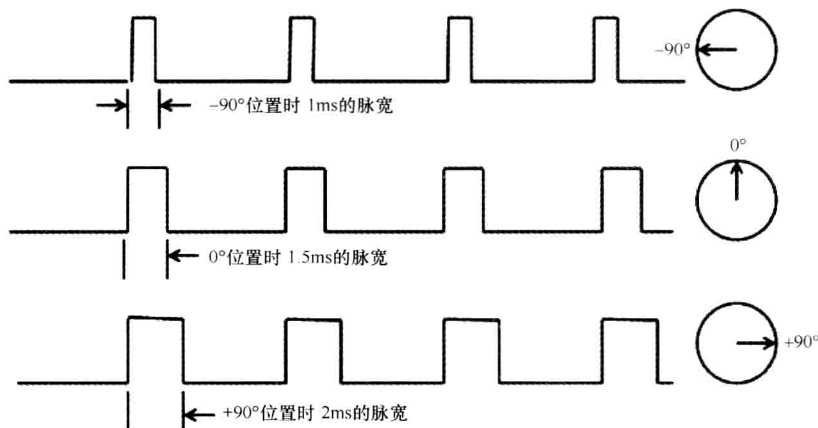


图 2.11 PIC 16F876A 处理器在 RB4、RB5、RB6 和 RB7 引脚处产生的输出波形

Pulse width of 1ms for -90 position -90 位置时 1ms 的脉宽

如上所述，在该具体实现中，采用了 SPI 模式而不是 I^2C 模式来进行串口通信。众所周知，I^2C 模式更适用于多分支总线架构的设备，而系统也随着所涉及的复杂性增加而变得更复杂。另一方面，SPI 模式适合于单分支、点对点的架构，系统也相对简单。因此，SPI 模式更适合于所开发的系统，因为该系统只需与 KOALA 机器人的点对点通信，这有助于使得系统相对简单，尤其是对于实际通信的情况。

2.3 KOALA 机器人与 PC 和视觉系统构成的集成系统

图 2.12 给出了基于 PC 的 KOALA 机器人通过 PIC 16F876A 微控制器板与视觉系统中 RC 伺服电动机进行通信的一个完整集成系统。PIC 微控制器与四个外设(即视觉系统中的四个伺服电动机)进行通信。这四个伺服电动机用于视觉系统的云台控制、左/右摄像头的聚焦控制。视觉系统所增加的四个自由度及其有效控制提高了整个集成系统的灵活性和自动化程度。

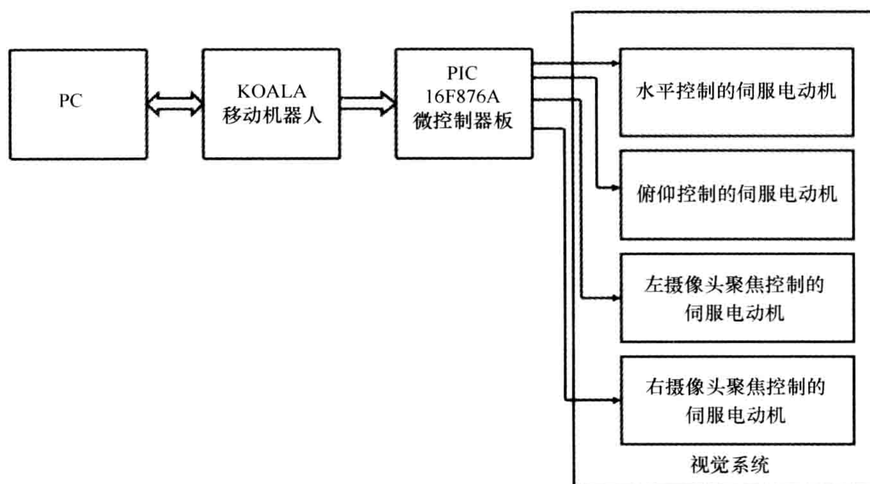


图 2.12 由 KOALA 机器人与 PC 和视觉系统构成的集成系统

另外还开发了一个基于 PC 的系统来与 KOALA 移动机器人通信。这是一个通过发送 ASCII 码字符串命令与机器人通信的 GUI 系统，同时也可接收机器人返回的传感器读数。图 2.13 给出了与用户交互的开发界面。其中，向用户提供了在串行传输模式下驱动命令的发送按钮。当点击“Send”按钮时，GUI 系统以串行模式传输 ASCII 码字符串，同时还在不同的显示框中显示从机器人处接收的数据。这些显示框有助于检查系统是否以预期方式执行命令任务。

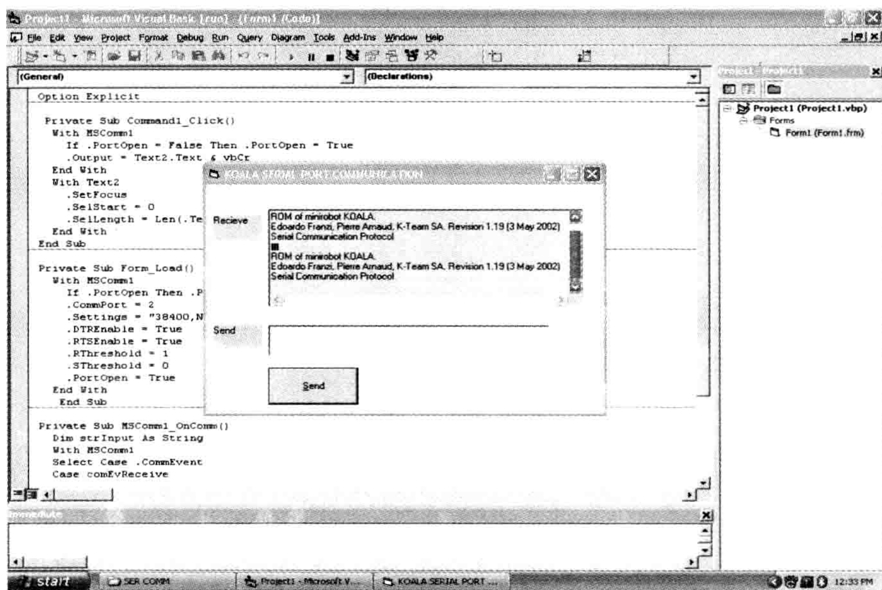


图 2.13 与用户交互的开发界面

在 KOALA 机器人终端，开发了另一个 C 程序，并将其交叉编译程序(扩展名为 . s37)下载到 Motorola 处理器中。该 . s37 程序以一种总是需要一个驱动命令的中断模式与 PC 中的 VB 程序进行通信，即发送 ASCII 码形式的消息，并以一种高度复杂的形式来响应该驱动命令。对于该串行通信模式，本地 PC 作为主机而 KOALA 机器人作为从机。主机 PC 与配置为远程终端单元(RTU)的 KOALA 机器人之间的每次交互都以如下方式进行[8]：

1）由主机 PC 发送给机器人一个以回车符(CR)结束的用户定义 ASCII 码字符串。

2）如果主机 PC 命令机器人采集和发送某个传感器读数(如红外传感器、声纳传感器等)，则 KOALA 机器人以一个回车符结束的 ASCII 码消息形式返回传感器读数。

图 2. 14 给出了一个控制 KOALA 机器人转向的简单 C 程序示例。图 2. 15 给出了在 PC 中开发的程序可通过交叉编译产生一个 . s37 文件，并下载到 KOALA 机器人中 Motorola 处理器的操作顺序。

```
#include <sys/Kos.h>
int
main ()
{
            mot_reset();
            mot_config_speed_1m (0, 1000, 800, 100);
            mot_config_speed_1m (1, 1000, 800, 100);
            mot_new_speed_1m (0, -6);
            mot_new_speed_1m (1, 6);
            return 0;
}
```

图 2. 14 控制 KOALA 机器人转向的一个 C 程序示例

在 KOALA 机器人的基本套件中已包含一组利用串行通信协议集在主机 PC 中打开一个终端仿真器并执行协议命令的工具和命令协议。这些协议命令包括一些非常有用的指令，如设置速度(“D”)、读取速度(“E”)、读取 A-D 转换输入(“I”)、读取管理传感器(“M”)和读取接近传感器(“N”)等。然而，除了这些 KOALA 机器人支持的指令外，还创建了一些与集成外设进行通信的协议指令。因此，创建了一套完整的协议指令集，使得主机 PC 以一种统一方式来控制机器人。这些指令包括与 KOALA 机器人所支持指令等效的指令，以及在本系统中创建的用于与附加外设进行通信的其他指令。另外，值得注意的是，当在 PC

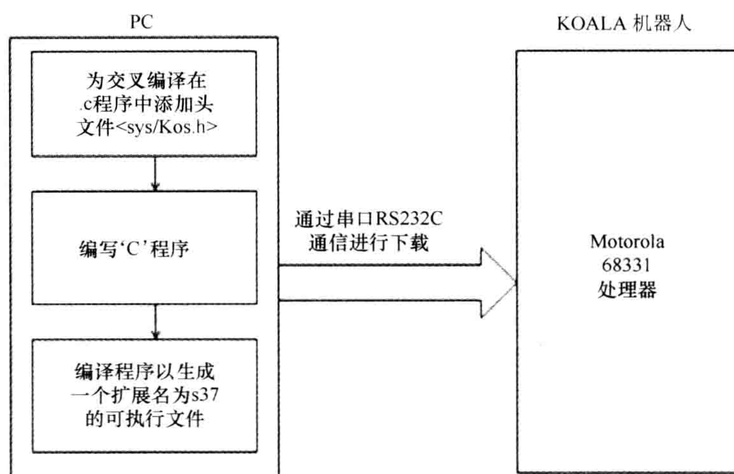


图 2.15　下载一个交叉编译 .c 程序的顺序操作的框图

的终端仿真器中执行 KOALA 机器人支持的指令时，首先应进入机器人处理器的监控程序模块并执行其设计功能。然而，当一个 .s37 程序下载到 Motorola 处理器中与 PC 端和 PIC 微控制器端进行通信时，将无法从终端仿真器激活监控程序。因此，将所有机器人协议指令处于同一框架(即从 VB 程序激活并由 .s37 程序执行)下是非常重要的。因此，为 KOALA 机器人编写的 C 程序必须包含所有由 PC 端发送的 ASCII 码请求消息。一组指令的示例列表以及主机发送的 ASCII 码消息形式见表 2.1。

表 2.1　主机 PC 和 KOALA 机器人之间进行串口通信的命令协议

命令性质	命令协议(ASCII 码字符串消息)	命令类别
设置 KOALA 机器人的电动机速度	<D><s><ddd><s><ddd><CR>	KOALA 机器人的等效指令
设置 KOALA 机器人的到达位置	<C><s><ddddd><s><ddddd><CR>	执行
请求发送红外接近传感器数据	<N><CR>	
请求发送红外环境光传感器数据	<O><CR>	执行
请求发送声纳传感器数据	<I><CR>	执行
请求发送电动机速度信息	<E><CR>	执行
设置电动机位置	<G><CR>	执行
请求发送电动机位置信息	<H><CR>	执行
设置 RC 伺服电动机的位置命令	<Z><i><s><dd><CR>	附加的外设接口命令

在表 2.1 中，例如要设置 KOALA 机器人的电动机速度，可以首先是标识符'D'，然后是由一个三位数字设置的每台电动机的速度(KOALA 机器人是一个差动驱动系统)。其中，先是'Motor 0'的速度指令，然后是'Motor 1'的速度指令。由于 KOALA 机器人中每台电动机速度的极性可分别设置为前向运动和后向运动，因此在各台电动机的速度信息之前增加一个符号位，在指令协议中表示为

<s>。设‘+’表示前向运动，‘-’表示后向运动。当 Motorola 处理器接收到这个 ASCII 码字符串消息后，.s37 程序就开始进行字符串解码并根据指令协议通过驱动 KOALA 机器人的每个电动机来执行该指令。执行每个 KOALA 机器人等效指令(作为一个 ASCII 码字符串消息由 PC 发送)时也是类似的动作。

然而，当 Motorola 处理器接收到附加外设接口命令相应的 ASCII 码字符串消息时，将正确地解码协议指令，并向 PIC 处理器发送合适的驱动指令。在这种情况下，如图 2.7 所示，Motorola 处理器工作在主模式下，而 PIC 处理器工作在从模式下。对于由 PC 发出的该行为的字符串消息，首先是标识符‘Z’，接下来是位置命令所对应的伺服电动机 ID <i>(i 从 1 变化到 4)，然后是位置命令的符号位 <s>(‘+’或‘-’)，最后是单位为度的表示实际位置命令的两位数字(-90°~90°)。每个 ASCII 字符串消息都以回车符 <CR> 结束。一旦 Motorola 处理器接收到该字符串，就对该字符串进行解码来产生电动机 ID 的一个字节信息，以及相应角位置命令的一个字信息(0°对应 1500μs，-90°对应 1000μs，+90°对应 2000μs，每个中间角位置命令对应成比例的时间内插值)，然后该字信息进行分解以分别产生相应的高字节信息和低字节信息。

2.4 实际性能评估

通过从主机 PC 端(即从 VB 程序端)串行通信向 KOALA 机器人发送 ASCII 码消息指令来对所有协议指令进行性能评估。在进行评估实验之前，分别单独测试对每个 RC 伺服电动机的校准，并引入软件校正来补偿小的偏移误差。进行了一系列实验来测试机器人集成系统实际执行 ASCII 码字符串消息所发布的任务的情况，其中一些实验情况描述如下。

在一个实验中，为测试驱动运动，要求机器人以某一较小的均匀速度前向运动。图 2.16 给出了 PC 上 GUI 系统的程序界面，这时发送的 ASCII 码字符串消息为“D+002+002”。可看出机器人运动与所执行的命令一致。同理，在较高运动速度指令、反向运动指令以及电动机差动驱动指令下来测试这些指令，同样可看出该机器人集成系统的行为与各个指令一致。

接下来，进行另一组实验，主要测试主机 PC 与 KOALA 机器人之间的双向串行通信。在这些实验中，要求机器人读取传感器读数，并将其返回给主机 PC。PC 端发出 ASCII 指令‘N’来读取安装在机器人周围所有 16 个红外接近传感器的读数。这些红外传感器的测距范围是 5~20cm。由红外传感器测量的每个输出值都是一个经 10 位 A-D 转换后的模拟量。在红外传感器 L0 和 R0(机器人轴线方向上放置的两个红外传感器)前方分别距离 5cm 和 20cm 处放置一个 5cm 宽的障碍物。实验情况与实际情况相符，距离较近时，返回的传感器读数较大。当距

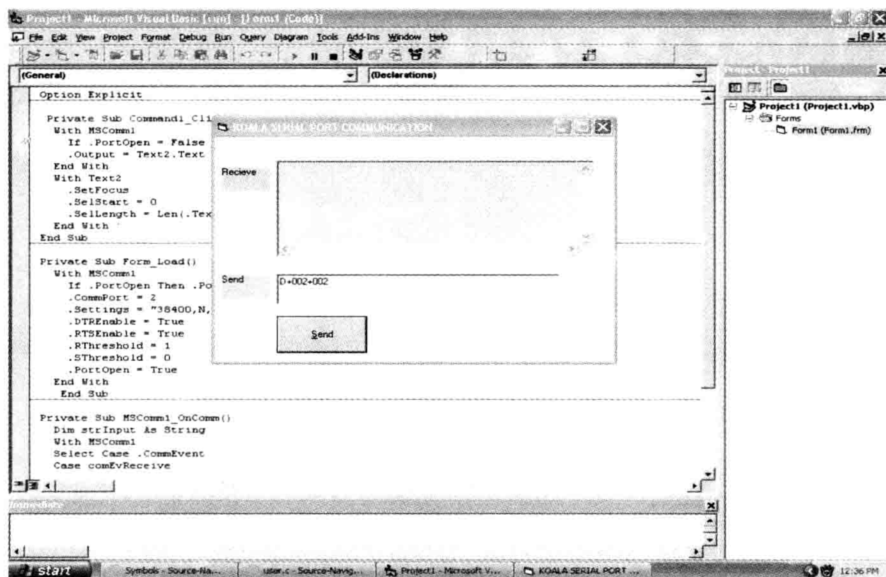
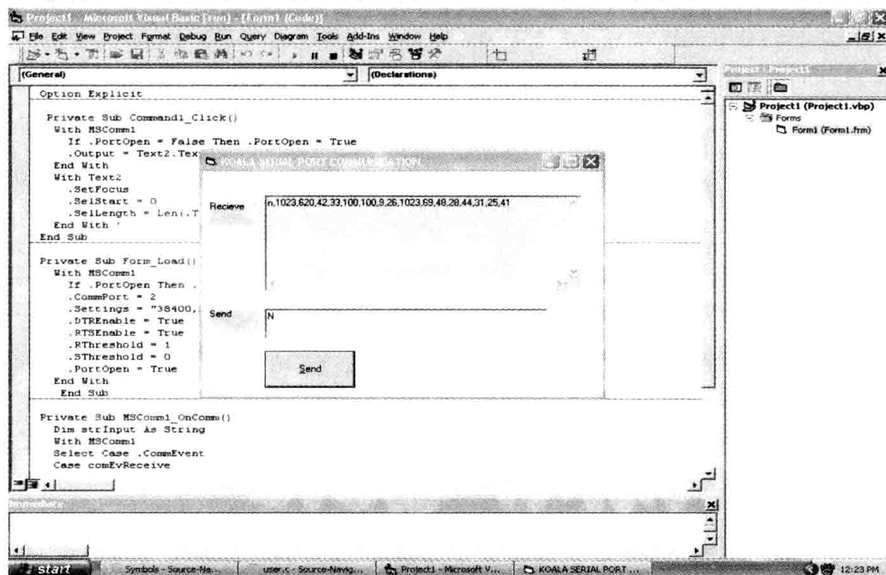


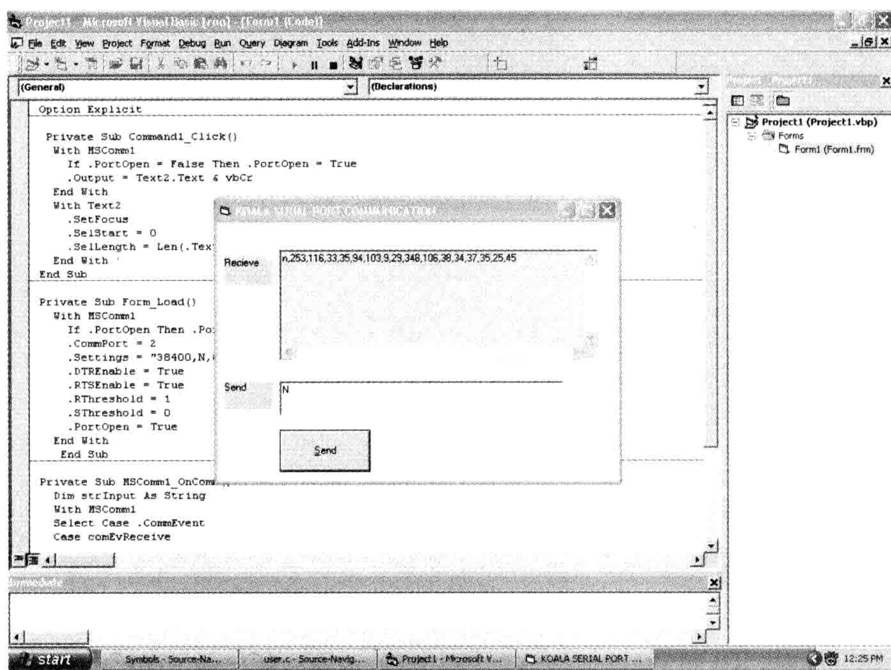
图 2.16　对机器人集成系统执行前向驱动命令时的 GUI 程序界面

离只有 5cm 时，L0 和 R0 返回的读数为 1023，这是可能的最大值。其他红外传感器的读数也以字符串形式获得，并在主机 PC 端显示，如图 2.17a 所示。由图



a)

图 2.17　对于一个 5cm 宽的障碍物，机器人集成系统分别在距离 5cm 和 20cm 处执行读取红外传感器数据命令时的 GUI 程序界面

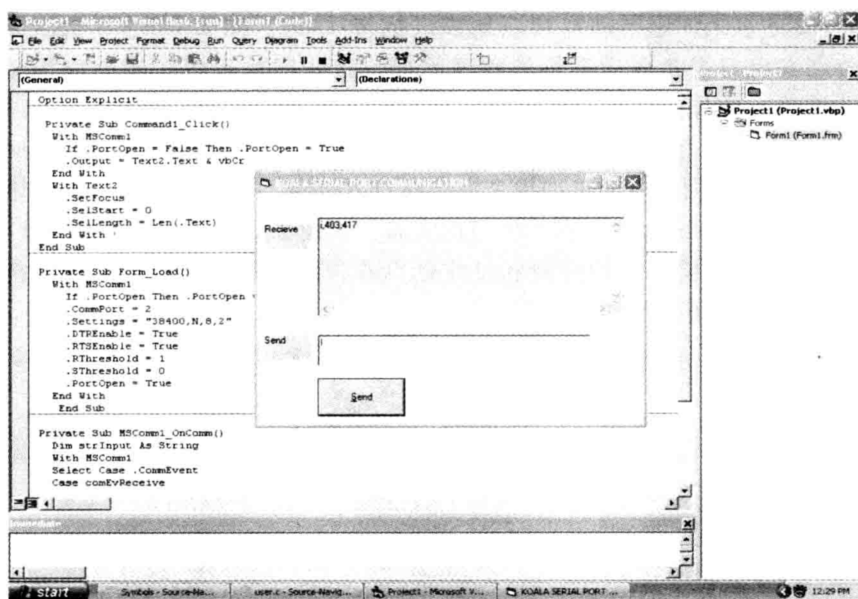


b)

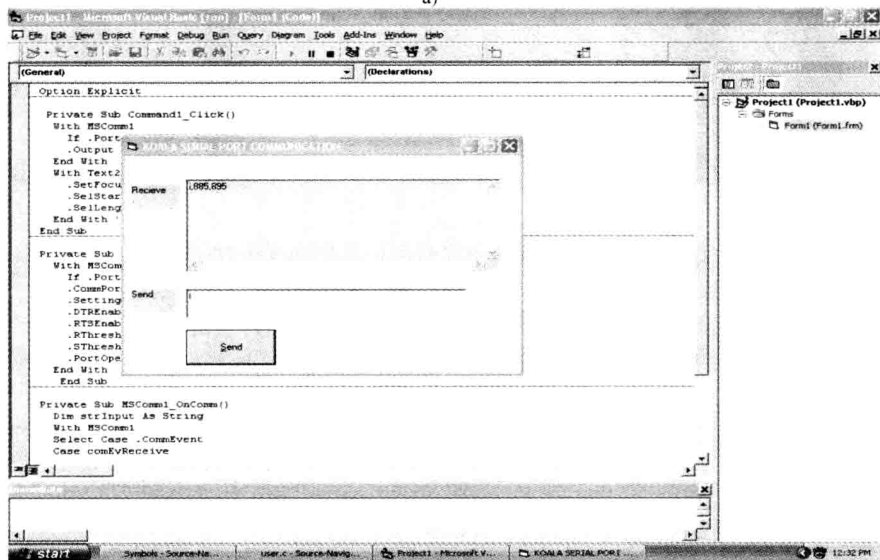
图 2.17　对于一个 5cm 宽的障碍物，机器人集成系统分别在距离 5cm 和 20cm 处执行读取红外传感器数据命令时的 GUI 程序界面(续)

可知，对于其他红外传感器，在给定的障碍物位置下，由于每个传感器与障碍物之间的距离大于 L0 和 R0 与障碍物之间的距离，因此传感器读数较小。若障碍物与机器人朝向之间的距离大于 20cm，则 L0 和 R0 所读取的读数分别为 253 和 348，如图 2.17b 所示。相应地，从其他传感器获得的读数也显著减小。同理，由 PC 端发出 ASCII 指令‘I’可获得两个声纳传感器的读数。这些声纳传感器可用于检测距离 15～300cm 的障碍物。相应的模拟输出量为 0～4.096V。对应的数字量为 0～1023，其中 0 对应 0V(最小距离)，而 1023 对应 4.096V(最大距离)。机器人分别位于距离墙面 100cm 和 200cm 处所获得的读数分别如图 2.18 所示。当与墙距离 100cm 时，返回的声纳读数分别为 403 和 417，而当与墙距离 200cm 时，返回的声纳读数要大得多，分别是 885 和 895。

接下来，进行的另一组重要实验是附加外设的接口指令，这时，主机 PC 端(工作于主模式)发出的串口指令以 ASCII 标识符‘Z’开始。该指令由从模式下 KOALA 机器人中的 Motorola 处理器接收，随后 Motorola 处理器(这时工作于主模式)对每个伺服电动机发出指令，通过 PIC 微控制器板来初始化串行通信，其中 PIC 16F876A 微控制器工作于从模式。图 2.19a 给出了每个伺服电动机位于中间



a)



b)

图 2.18　机器人集成系统分别在距离墙 100cm 和 200cm 处执行读取声纳传感器数据命令时的 GUI 程序界面

位置时的初始状态。然后，分别依次发出四个指令‘Z1 +45’，‘Z2 +45’，‘Z3 +45’和‘Z4 +45’，使得用于左摄像头聚焦控制、右摄像头聚焦控制、水平控制

和俯仰控制的每个伺服电动机相应地依次旋转 +45°。图 2.19b ~ e 给出了主机 PC 端分别发出每个指令后系统的程序界面。这些实验表明在实际应用中，与 KOALA 机器人集成的附加外设可成功地与机器人接口，并按预期从主机 PC 端获得合适的指令。表 2.2 给出了从 PC 端(在此用户发出指令)发出指令的时间延迟，以及每个 RC 伺服电动机按指令完全执行其功能。对于每个 RC 伺服电动机分别进行角位置指令为 45°和 90°的两次实验。由表 2.2 可知，左摄像头聚焦轴、右摄像头聚焦轴和偏转轴的感应和反应延迟非常接近。而相对于其余三个自由度，水平轴在实际中的时延较大。

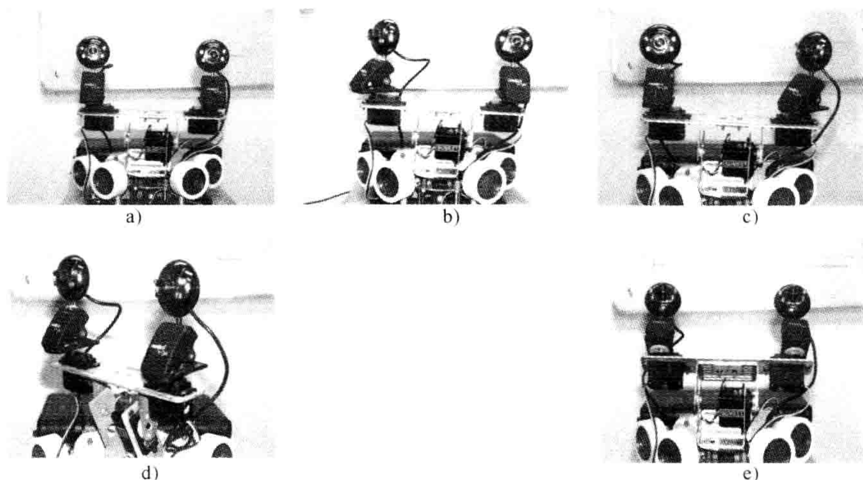


图 2.19 初始化条件下[即每个伺服电动机均位于 0°角度位置(称为中心位置)]的系统配置图片(见图 2.19a)和由 PC 依次发送的对于四个分别进行左摄像头聚焦控制、右摄像头聚焦控制、水平控制和俯仰控制的 RC 伺服电动机执行 45°角位置指令时的系统位置图片(见图 2.19b ~ e)

表 2.2 RC 伺服电动机感应和反应延迟时间

自由度	发送的角位置指令/(°)	感应和反应延迟时间/s
左摄像头聚焦轴	90	0.433
	45	0.200
右摄像头聚焦轴	90	0.433
	45	0.200
水平轴	90	0.633
	45	0.333
偏转轴	90	0.400
	45	0.300

2.5 本章小结

本章详细介绍和阐述了如何开发一个基于 PIC 微控制器的系统用于与现有移动机器人(在此是 KOALA 机器人)的实际外设接口。在所开发的系统中，以中断驱动模式进行串行通信，其中 KOALA 机器人的处理器工作于主模式，而 PIC 处理器工作于从模式。在实验室环境下开发了一个由 PC、KOALA 机器人、PIC 微控制器板和基于双摄像头的视觉系统组成的完整集成系统。其中，PIC 微控制器系统是用于控制视觉系统中四个自由度的四个外设(即四个 PC 伺服电动机)。该完整集成系统在 PC 的 GUI 系统控制下工作，其中 PC 端的系统工作在主模式，并以中断驱动模式与工作于从模式的 KOALA 机器人中的 Motorola 处理器进行串行通信。

所开发的系统表明如何在机器人基本平台的外部集成附加外设来提高集成系统的能力和复杂性，且开发成本低。这样的思路也可扩展到增加更多的其他外设，同时也有助于在其他实际移动机器人上实现。另外，也希望本章中对该系统的详细介绍能有助于机器人领域的研究人员来开发类似的集成系统。由于通常情况下，一个机器人集成平台中的集成传感器和其他部件的技术说明书都只由生产厂商掌握，而无法公开获得，因此这将对开展机器人研究非常有用。同时，还应注意到，尽管在此所给出的所有实验都是基于双摄像头系统进行的，但该系统同样也适用于单摄像头系统，这时将摄像头放置在主动式云台系统的中央位置，这种情况下，只能控制两个自由度(仅水平控制和俯仰控制)。

致谢　本章介绍的研究工作得到了印度大学教育资助委员会重大研究项目计划的资助(资助号：32-118/2006(SR)]。

参考文献

[1] Turkoglu, I.: Hardware implementation of varicap diode's ANN model using PIC microcontrollers. Sensors and Actuators A: Physical 138, 288–293 (2007)

[2] Frankowiak, M.R., Grosvenor, R.I., Prickett, P.W.: A Petri-net based distributed monitoring system using PIC microcontrollers. Microprocessors and Microsystems 29, 189–196 (2005)

[3] O'Neill, R.W., Greenwood, J.B., Gradziel, M.L., Williams, D.: Microcontroller based double beam modulation system for atomic scattering experiments. Measurement Science and Technology 12, 1480–1485 (2001)

[4] Hung, D.L., Wang, J.: Digital hardware realization of a recurrent neural network for solving the assignment problem. Neurocomputing 51, 447–461 (2003)

[5] Colak, I., Bayindir, R., Sefa, I.: Experimental study on reactive power compensation using a fuzzy logic controlled synchronous motor. Energy Conversion Management 45(15-16), 2371–2391 (2004)
[6] PIC 16F876A Data sheet, 28/40/44- Pin Enhanced Flash Microcontrollers. Microchip Technology Inc. (2003)
[7] HS-322, H.R.: Servo Motor manual, HITEC RCD KOREA Inc. (2002)
[8] KOALA User Manual, Version 2.0(silver edition). K-team S.A., Switzerland (2001)
[9] MC68331 User Manual, MOTOROLA, INC. (1996)
[10] Nirmal Singh, N., Chatterjee, A., Rakshit, A.: A PIC Microcontroller system for Real-Life Interfacing of External Peripherals with a Mobile Robot. International Journal of Electronics 97(2), 139–161 (2010)
[11] Nirmal Singh, N.: Vision Based Autonomous Navigation of Mobile Robots. Ph.D. Thesis, Jadavpur University, Kolkata, India (2010)

第 3 章　利用子目标的基于视觉的移动机器人导航㊀

摘要： 本章介绍了如何开发一个两层架构且与红外传感器相结合的基于视觉的机器人导航机制。该算法采用一种基于子目标的机制来通过跟踪最短路径以达到最终目标，并同时实现预期的避障功能。以创建下一个子目标的形式迭代运行该算法，并在单次迭代中导航到该点，从而尽可能地在最少迭代次数下达到最终目标。

3.1　简介

机器人领域的一些最新技术发展对许多工业和社会领域做出了重大贡献。目前，在工厂自动化、监控系统、质量控制系统、自主导引车(AGV)、防灾和医疗辅助等领域都有着机器人系统的大量应用。现在，越来越多的机器人应用旨在提高人类日常生活，并且机器人也正前所未有地执行各种不同任务[1]。对于许多这样的应用，机器人的自主移动能力是其中一个关键因素。许多现代机器人的应用中，都采用计算机视觉作为主要的感知手段。如前所述，视觉系统可看作一种被动传感器，它比红外、激光和声纳等主动传感器具有更多的优点。被动传感器(如摄像头)在采集数据过程中发射光或波不会改变环境，并且在所获得的图像中包含比主动传感器更多的信息(如物质、空间和时间信息)[2]。视觉是能够让人类提取客观物理世界相关信息的一种感知，同时也是人类最依赖的感知。能够提取这些信息的计算机视觉技术不断发展，现今移动机器人已实现了越来越多的基于视觉的实时导航系统。

基于视觉的机器人导航是一种能够引导移动机器人到达预期位置，或沿环境中一条期望路径运动，并主要利用视觉传感器避开静态障碍物(或许是动态障碍物)的技术[3,4]。本章将介绍一种移动机器人导航机制的具体实现，其中视觉感知作为路径规划的主要传感器，而红外传感器作为在一个静态或动态室内环境中具有避障功能的移动机器人实际导航的辅助传感器。如前所述，构建环境

㊀ 本章是基于 N. Nirmal Singh，Avishek Chatterjee，Amitava Chatterjee 和 Anjan Rakshit 2011 年 5 月发表于 Measurement 期刊第 44 卷，第 4 期，620 ~ 641 页的论文“A two-layered subgoal based mobile robot navigation algorithm with vision system and IR sensors”。经 Elsevier 出版商许可后重印。

地图的常用方法基于栅格法[5-7]、拓扑地图[8,9]和混合地图[10]等。无地图导航系统是指在整个导航空间中无需显式表示，而是通过识别环境中的物体，并根据视觉观测产生运动命令来跟踪或避开这些物体的导航系统[11,12]。另外还有一些研究工作提出利用立体视觉[13,14]、三目视觉[15]、全方位或全景视觉[16,17]和单目视觉[18,19]来获取环境信息。移动机器人导航中的每种方法都有各自的优点和缺点。在地图知识已知的情况下，导航的一个重要问题是用于移动机器人智能控制或引导的路径规划问题。路径规划的常用方法是基于地图法、栅格分解法和势场法等方法[20]，不同之处在于如何构建连通图及其表示。显然，如果没有环境的任何先验知识，就几乎不可能在所有可能路径中确定最短的导航路径。确定最短路径可采用标准的图搜索技术，如 Dijkstra 算法[21]和 A * 算法[22]。

如上所述，本章将介绍一种目标驱动方法来进行移动机器人导航，即采用视觉感知和红外导航[28,29]。在一个存在障碍物的静态或动态环境中，这种基于两层的方法可确定起始点和已知目标点之间的最短导航路径。在第一层中，视觉作为主要感知系统为随后的路径规划采集环境图像。首先，对所采集的图像执行一系列的图像处理操作，然后采用梯度下降算法来计算机器人当前位置与目标点之间的最短路径，并实现避障[26]。利用该最短路径来产生子目标，然后在基于红外传感器的导引基础上，利用这些子目标信息来进行局部导航。即使在环境动态变化的情况下，第二层中基于红外传感器的机器人导航也能够导引机器人到达子目标。一旦机器人达到子目标，就再次激活该两层算法来产生一个新的子目标，并将机器人导航到新的子目标。重复执行该过程，直到达到最终目标。该方法想要同时实现两个目标。首先，基于视觉感知，要力图实现一个最短路径规划算法，以尽可能快地在避障的同时达到目标点。然后，如果在导航过程中环境发生变化且障碍物信息更新，则基于红外传感器的导引将使得机器人具有应对环境变化的能力，从而保证机器人仍然能够安全导航。定期利用视觉来更新环境信息，那么随后的路径规划和红外导航就能够帮助机器人在导航过程中适应环境的动态变化，并在尽可能短的时间内到达目标点。本实验室开发的算法可对 KOALA 机器人创建几种实际环境[23]。实验结果表明了所提算法的有效性。该算法将在本章随后的内容中详细介绍。

3.2 硬件设置

第 2 章已详细介绍了 KOALA 机器人。在此，再次简要概述一下 KOALA 机器人的突出特点，并简单介绍在实际实现中所用的硬件设置。KOALA 机器人是一个由瑞士 K-team 公司生产的六轮差动驱动小车（32cm × 32cm）[23]。本实验室所用的 KOALA 机器人配置有 16 个接近/环境红外传感器、四个声纳传感器和车

轮编码器，并在加尔各答贾达普(Jadavpur)大学电子工程系的电子测量与仪器实验室中对该机器人集成了两个完整视觉系统。所开发的视觉系统既可用作采用双摄像头的立体视觉系统(如第 2 章所述)，也可用作采用单摄像头的视觉系统。现在，介绍一种采用一个无线摄像头作为单目视觉的算法。在 KOALA 机器人中，利用一个车载微控制器(Motorola 68331，工作频率为 22MHz)来进行硬件控制[23]。图 3.1a 给出了实验室所采用的集成了四个声纳传感器和由单目视觉构成的视觉系统的移动机器人图片。其中，声纳传感器可检测 15 ~ 300cm 范围内的障碍，而红外传感器的测量范围为 5 ~ 20cm。本系统采用的单目视觉系统是由一

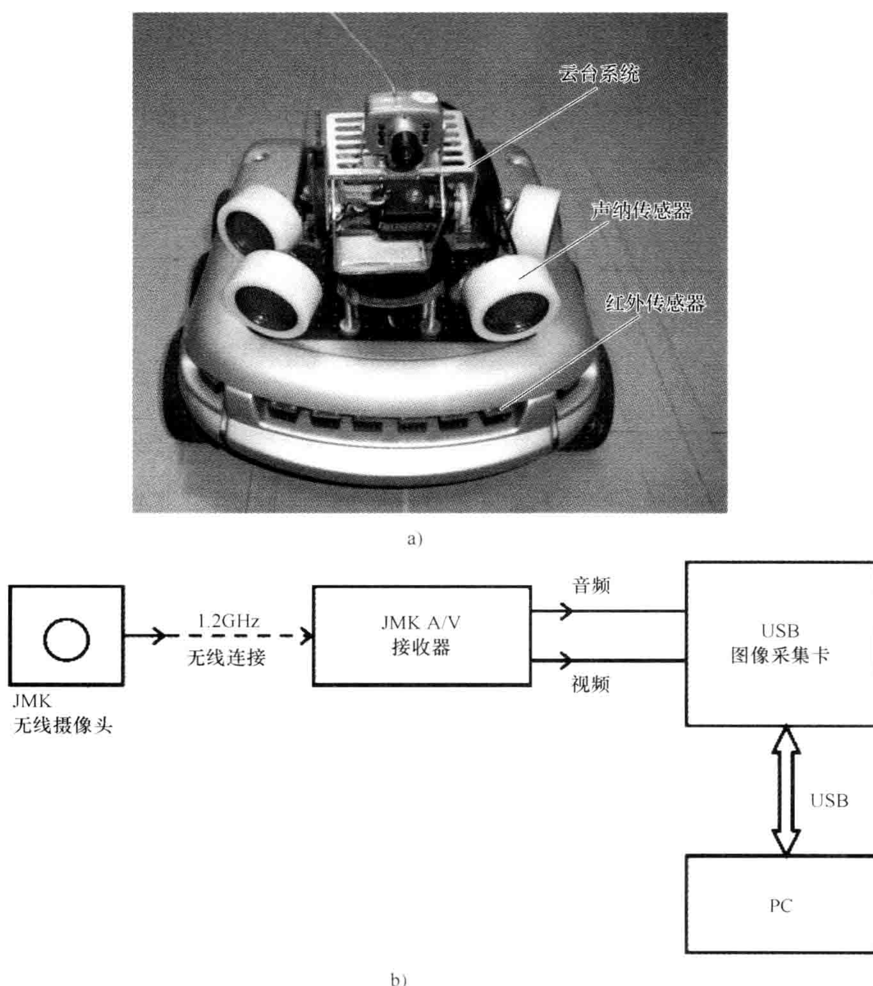


图 3.1　配置有声纳和红外传感器并集成了基于单摄像头的视觉系统的 KOALA 机器人(见图 3.1a)以及视觉系统的框图(见图 3.1b)

个具有 A/V 接收器的 JMK 无线摄像头(WS-309AS)和一个可采集连续视频流的 Frontech USB 图像采集卡(TV Box)组成的。图 3.1b 给出了该视觉系统的原理示意图。整个系统开发的目标是提供一个在工业领域具有吸引力的低成本解决方案，所开发的单目视觉系统应具有可进行云台控制的两个自由度。为对该视觉系统增加两个自由度，本实验室开发了一个 PIC(16F876A)微控制器系统来对基于单目摄像头的机器人系统进行云台控制[24]。在此，车载 Motorola 微控制器作为主机，而 PIC 微控制器作为从机。以中断驱动模式开发的软件可实现通过 RS232C 与移动机器人进行通信。图 3.2 给出了 PC 端开发的可与用户交互的用户接口程序界面。主要的串口通信是由 PC 与机器人 Motorola 处理器之间的 ASCII 码消息来控制的。

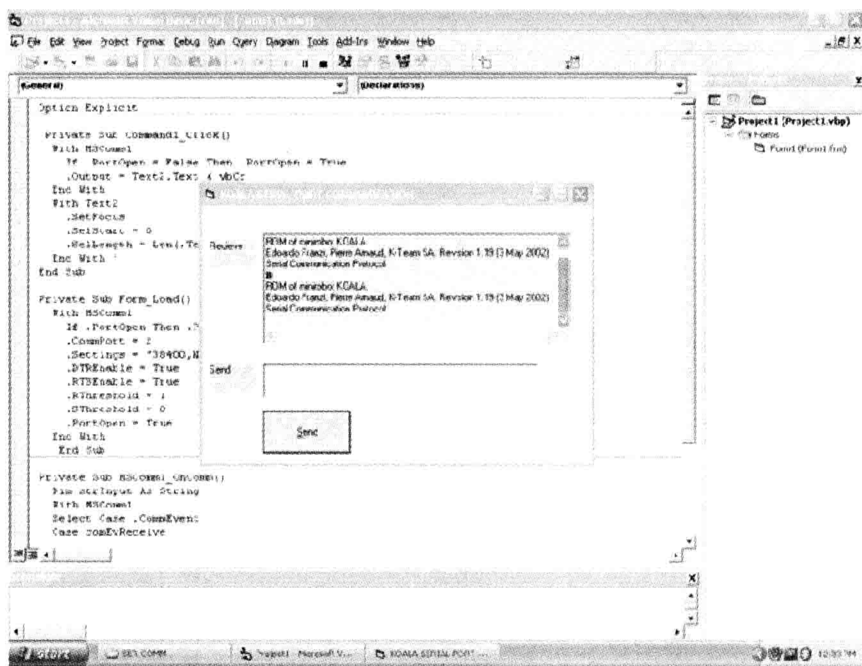


图 3.2 可与用户交互的用户接口程序界面

PC 与机器人之间的 RS232C 串口连接通常设置为一个起始位、两个停止位且无奇偶校验位模式的 8bit 数据。例如，由 PC 端发送的用于云台控制的 RC 伺服动作的字符串消息中包括一个标识符‘Z’，然后是位置命令相应的伺服电动机 ID <i>(i=1，2)，接下来是位置命令的符号位 <s>(‘+’或‘-’)，最后是两个实际的位置命令位，单位是度(可从 -90°变化到 +90°)。通常，由回车符 <CR>来结束每个 ASCII 码字符串消息。

3.3 目标导向的两层导航机制

图 3.3 给出了所提出的完整导航算法的流程。如图 3.1a 所示的一个无线摄像头用于采集 KOALA 机器人前方环境的连续视频流。对该视频流中任意时刻的一帧采集图像进一步处理。首先对采集图像帧进行一系列图像处理操作，如图像滤波、边缘检测和图像分割等，使得图像适合于进一步处理。然后，根据所提供的先验目标信息，对处理后的图像采用最短路径生成算法。接下来，利用所产生的最短路径来确定下一个子目标。整个过程构成了算法的第一层，并在 PC 中利用 VB 平台高层实现。然后，将该子目标信息传给 KOALA 机器人利用避障功能朝子目标实际导航的算法第二层，从而实现机器人在一个动态变化环境

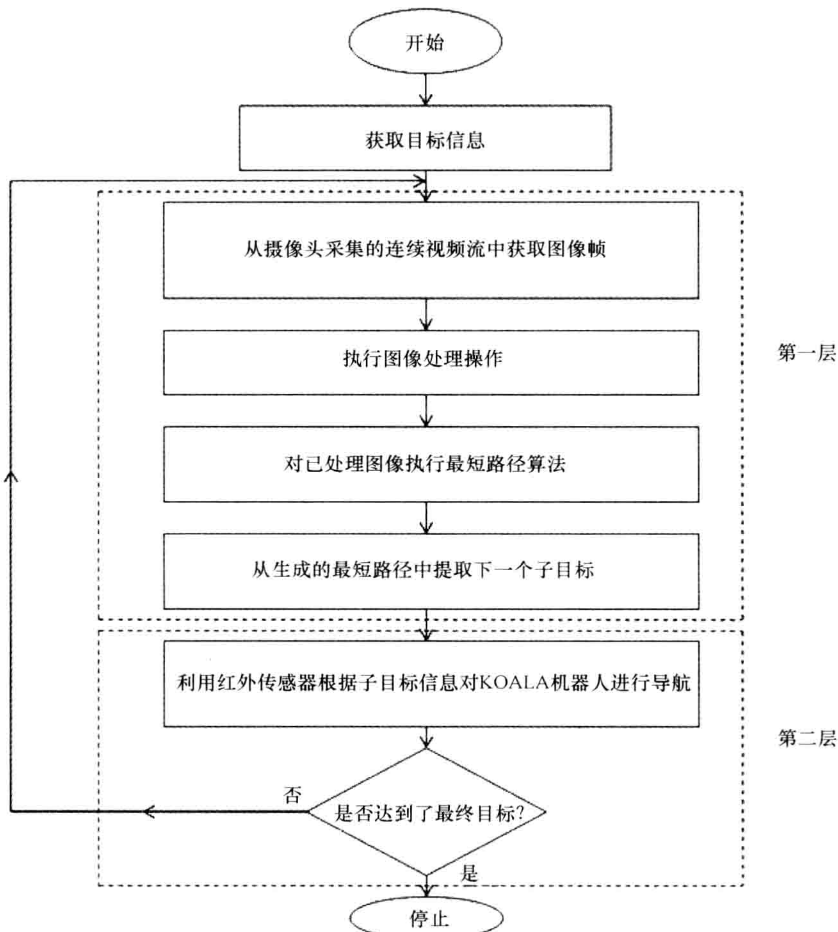


图 3.3　所提导航算法的流程

中的运动。第二层中的导航是利用几个安装在 KOALA 机器人前方和两侧的红外传感器来实现的。一旦机器人到达子目标，就又返回到第一层控制，从而产生下一个子目标，并在第二层中进行导航。迭代执行这种局部路径规划和实际导航的过程，直到达到最终目标。第二层中实际导航的算法是由一个下载到 KOALA 机器人中 Motorola 处理器的交叉编译 C 程序（一个 . s37 文件）实现的，该 . s37 文件与 PC 端的 VB 程序以中断驱动方式通信。由 C 程序生成的 . s37 程序还可对 PC 端的 VB 程序提供一组控制 KOALA 机器人的协议命令，这些命令原本只适用于在 KOALA 机器人套件提供的终端仿真器中执行。在此，开发了一个由 C 程序执行的支持所有 KOALA 机器人协议指令以及附加导航算法的系统，使得整个系统都能够完全由 PC 端的 VB 程序控制。

3.4 第一层中基于图像处理的环境探索

图像处理是一种以图像为输入信号，输出信号为输入图像变换形式的信号处理形式。本书所提出的系统采用一种基于图像分割的地图构建方法，用于室内环境下移动机器人的视觉导航，并假设表面一致。具体实现步骤如下[27]：

1. 从无线摄像头采集图像

安装在机器人云台系统中心的摄像头不断采集机器人前方环境的视频流。可从该视频流中采集一帧图像来进一步处理。图 3. 4a 给出了一帧采集图像。

2. 对采集图像进行低通滤波

接下来，对采集图像进行低通滤波以减少噪声。该处理可对邻近像素点产生平滑或模糊效果。本书所开发的系统采用常用的算术均值滤波来进行低通滤波，该算术均值滤波对每个像素点中心采用一个 5 ×5 矩阵，并将该滤波矩阵作用下的像素点平均值作为该像素点的亮度值。

3. 对滤波图像进行 Canny 边缘检测

边缘表示不同灰度值的两个区域之间的边界。基于边缘的分割技术是指利用不同类型的梯度算子从场景中分离出期望目标。在本系统中采用 Canny 边缘检测算法来检测图像中的边缘。图 3. 4b 给出了经低通滤波处理后采集图像的边缘图像。

4. 加粗并连接边缘图像中的边缘

边缘图像中包含许多小的杂散边缘。为使得边缘图像更有意义，需要将相邻边缘连接并加粗来使之更加明显。通过一种称为膨胀的形态学操作来进行边缘加粗处理，即利用一个可增长图像中选择区域的前景像素的结构元素。通常是对一幅二值图像采用膨胀操作，并将所产生的另一幅二值图像作为输出。该膨胀操作可“加粗”或“增长”二值图像中的目标，加粗的形状可通过选择合适的

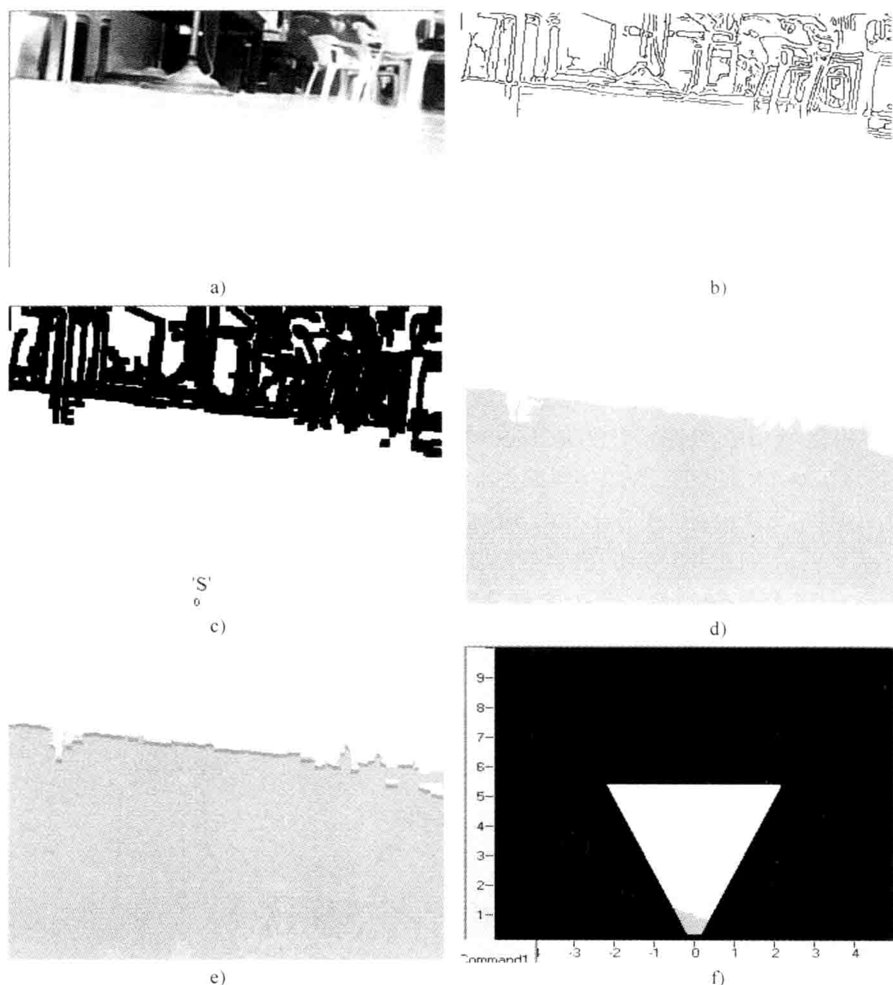


图 3.4　由无线摄像头采集的图像(见图 3.4a)、边缘检测图像(见图 3.4b)、边缘加粗的图像(见图 3.4c)、区域增长图像(见图 3.4d)、包含障碍物信息的图像(见图 3.4e)、梯形地面图像(见图 3.4f)

用于执行图像膨胀的结构元素的形状来控制。在边缘图像中连接并加粗边缘的思想还可通过选择合适的滤波掩码利用低通滤波来实现。在本系统中是通过几何均值滤波来实现该操作的。几何均值滤波器是一种非线性均值滤波器，在去除高斯噪声和保持边缘特征方面比算术均值滤波器更为有效。图 3.4c 给出了边缘连接和加粗后的图像。

5. 对边缘加粗的图像进行区域增长分割

一旦经过边缘加粗，图像就可分割为几个区域。为找出图像中的障碍区和非障碍区，对边缘加粗的图像执行基于区域增长的分割算法。区域增长算法是

一种简单而有效的基于区域的图像分割方法，可归类为一种涉及初始种子点选择的基于像素的图像分割方法。该分割方法首先检查初始“种子点”的邻近像素，来确定是否应将该像素点添加为种子点。然后，通过检查所创建的每个块的特性并根据某种准则（相似的像素灰度值、纹理等）将其与相邻块合并。选择图像底部中心附近的像素点作为进行区域增长的种子点，如图 3.4c 所示的点‘S’。现在，沿着从下到上的所有垂直线来扫描图像。在地面区域结束的点看作障碍物，而在障碍物之前的所有区域为自由区域，超过障碍物的所有区域看作隐藏区域。图 3.4d 中给出的下半部分表示非障碍区（自由空间），上半部分表示隐藏区。接下来，用两个区域之间的线来标记障碍区，图 3.4e 中包含了这三种区域。不断执行上述过程并更新障碍物信息。

6. 将区域增长图像转化为地面区域

将以障碍物信息更新的整个增长区域从像平面转换到地面区域。为利用单个摄像头来计算 3D 坐标系中的距离，在此假设所有障碍物都是与地面接触的，并在 2D 坐标下表示。图 3.5 表明在给定摄像头高度和俯仰角的条件下，像平面上的任意一点如何直接映射到相对于摄像头位置的地面上[25]。在此，假设机器人/摄像头的 3D 坐标系如图 3.5 所示。假设在任何时刻，该坐标系都与机器人/摄像头的当前位置关联。通过这种坐标变换的方法可在摄像头所采集的图像中确定自由点和障碍点在世界坐标系（WCS）中的坐标。由此，根据图 3.5，像平面中坐标为 (u, v)

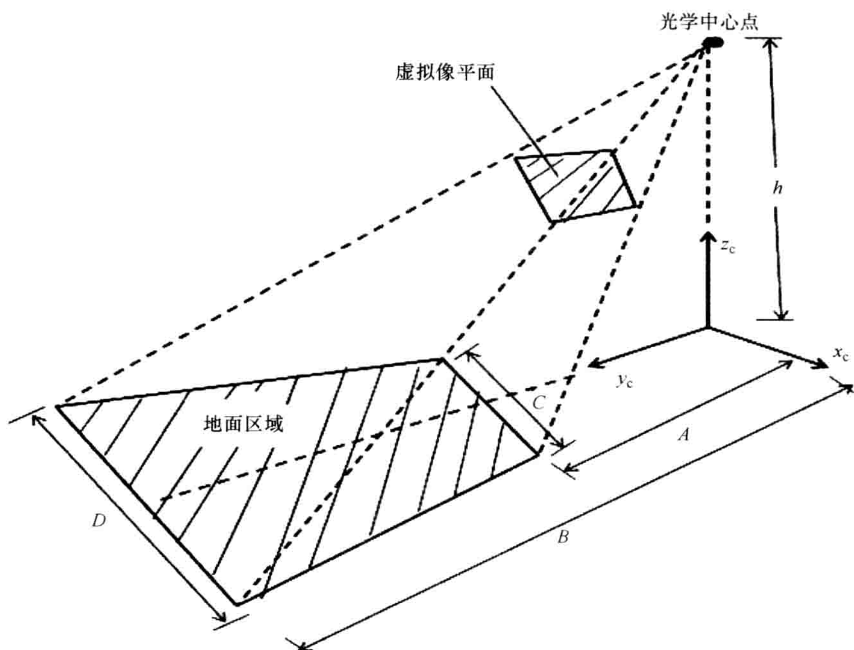


图 3.5 图像坐标与移动机器人坐标之间的关系图

的任意一点都可转换为机器人/摄像头坐标系中的二维坐标(x_c, y_c)，即

$$x_c = \frac{uh}{(f\sin\theta_{EL} + v\cos\theta_{EL})}$$

$$y_c = \frac{h(v\sin\theta_{EL} - f\cos\theta_{EL})}{(v\cos\theta_{EL} + f\sin\theta_{EL})}$$

式中　h——从基平面到摄像头光学中心的高度；

f——摄像头的焦距；

θ_{EL}——摄像头的俯仰角；

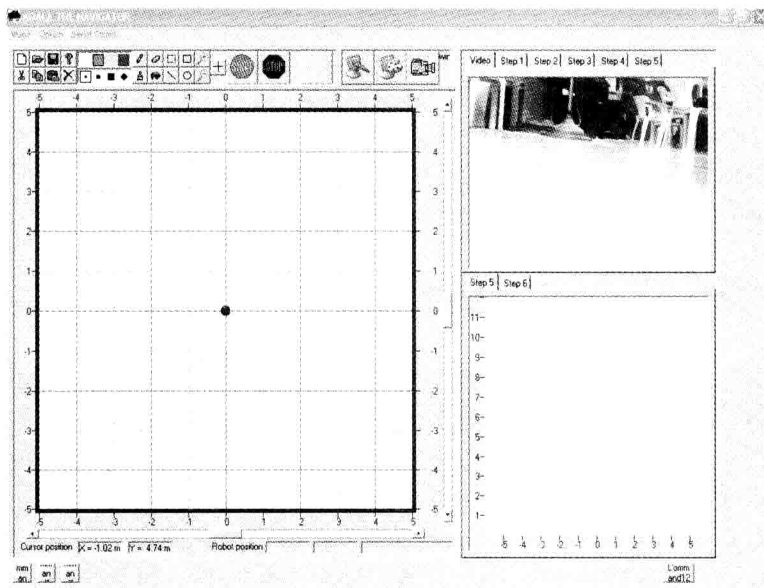
$y_{c_{min}} = A$(死区)；

$y_{c_{max}} = B$；

$x_{c_{min}} = \frac{C}{2}$；

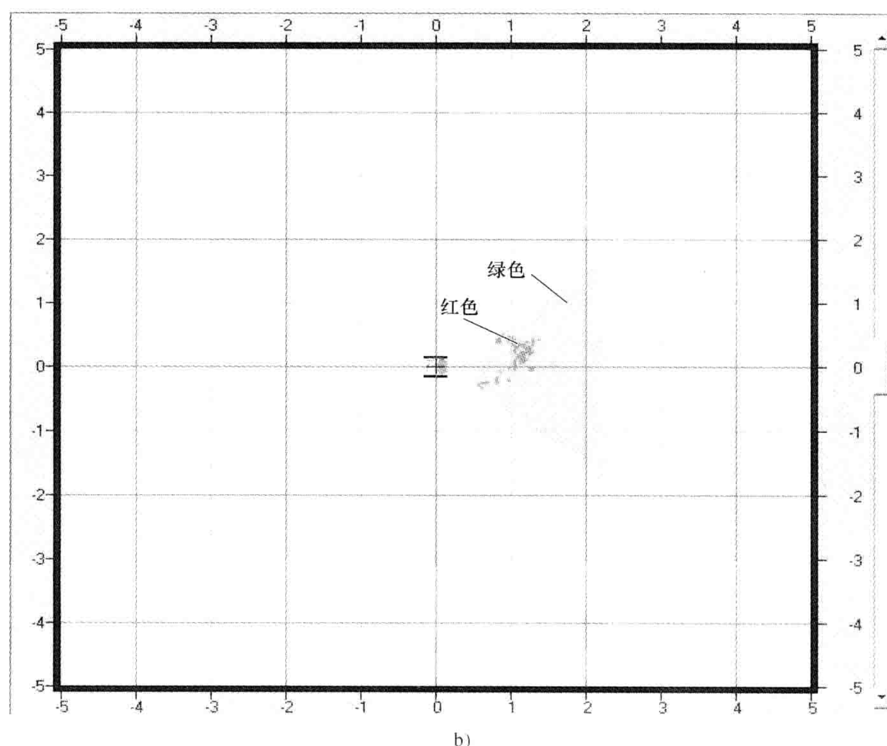
$x_{c_{max}} = \frac{D}{2}$。

一旦采用上述变换，在给定(x_c, y_c)和机器人当前位置(x_R, y_R, ϕ_R)的条件下，即可求得地面上点(x, y)的实际位置。图3.4f给出了具有障碍信息的地面，变换后的地面区域呈梯形状。然后，将该地面图像复制到一个500×500像素的地图(即20m×20m)中作为机器人的工作空间。图3.6a给出了地图创建的



a)

图3.6　地图创建软件开发界面(见图3.6a)
和包含障碍物信息的地图更新界面(见图3.6b)



b)

图 3.6 地图创建软件开发界面(见图 3.6a)
和包含障碍物信息的地图更新界面(见图 3.6b)(续)

界面，图 3.6b 给出了包含地面图像的栅格地图。在图 3.6b 中，梯形地面区域用绿色表示，障碍物信息用红色表示。即使在机器人运动时，也不断进行上述变换过程，并在地图中更新障碍物信息。

3.5 最短路径计算和子目标生成

在移动机器人导航过程中，确定机器人当前位置和目标点之间的最优路径是一个重要目标，这样就能使得机器人在避障的前提下尽可能短时间到达目的地。在此，采用一种基于启发式的梯度方法根据栅格地图来寻找最短路径[26]。算法 3.1 中详细描述了该算法。在之前的步骤中已知机器人的初始位置和目标位置以及障碍物信息，然后利用最速下降法来确定最短路径上的各点坐标。最速下降算法是采用梯度函数来确定函数下降最快的方向。以特定步长沿梯度函数下降最快方向迭代执行该算法，然后重新计算梯度来确定新的运动方向。如果地面上的障碍物以及初始点和目标点已知，那么这种启发式方法是很容易理解的。假设地面无任何摩擦，也就是说，如果在地面上的起始点处倾倒沙子，

那么这些沙子会向所有可能的路径扩散，这类似于流体在所有可能方向上的扩散。显而易见，肯定不会经过障碍。在每次迭代中，假设都倾倒固定数量的沙子并任其扩散。当沙子达到一定的预设高度时，对地面上的每个点都设置一个相当于迭代次数的时间序号。由此，越早达到该高度，则时间序号就越小。选择该预设高度和固定数量的沙子以避免在任意有限区域内饱和。由此，一个行程时间矩阵（$\boldsymbol{H}$）可通过有限元扩散法来计算，并迭代更新该 $\boldsymbol{H}$ 矩阵，直到满足终止条件。在上述过程结束时，$\boldsymbol{H}$ 矩阵中值为零的元素对应障碍单元。接下来，采用基于梯度下降的方法来确定从目标点开始到机器人当前位置的最短路径上各点的坐标。为此，计算 $\boldsymbol{H}$ 在 x 和 y 方向上的梯度矩阵，即 $\nabla \boldsymbol{H}_x$ 和 $\nabla \boldsymbol{H}_y$，并根据该梯度矩阵来计算最短路径上下一点的新坐标，直到求得最短路径上最后一点的坐标。该算法总是从目标点开始后向执行，这是一种迭代执行的有效方法。图3.7给出了一个计算出地图中初始点与目标点之间最短路径的示例环境。

算法3.1　可实现避障的最短路径生成算法

开始

1. 获取占据栅格矩阵（$\boldsymbol{M}$），起始点（x_start，y_start）以及目标点（x_goal，y_goal）。$\boldsymbol{M}(i, j)=0$ 表示一个自由栅格，$\boldsymbol{M}(i, j)=1$ 表示一个障碍栅格。

2. 创建扩散矩阵（$\boldsymbol{W}$）和行程时间矩阵（$\boldsymbol{H}$），并使这两个矩阵与 $\boldsymbol{M}$ 矩阵同样大小。初始化 $\boldsymbol{W}_0=\boldsymbol{H}_0=0$。

3. 设 $\boldsymbol{W}_0$(x_start，y_start) = 1。

4. 设置扩散系数（diffconst）和无更新的最大迭代次数（no_update_max）。初始化迭代次数（iter_count）和无更新的迭代次数（no_update_iter_count）。

5. WHILE（no_update_iter_count < no_update_max）

   5.1. iter_count = iter_count + 1。

   5.2. 向下扩展栅格：

   $$\boldsymbol{W}_{\text{iter_count}}(i,j)=\boldsymbol{W}_{\text{iter_count}}(i,j)+\text{diffconst}*\boldsymbol{W}_{\text{iter_count}}(i+1,j)$$
   $$i=1,2,\cdots,(\text{W_ROWS}-1);j=1,2,\cdots,\text{W_COLS};$$

   5.3. 向上扩展栅格：

   $$\boldsymbol{W}_{\text{iter_count}}(i,j)=\boldsymbol{W}_{\text{iter_count}}(i,j)+\text{diffconst}*\boldsymbol{W}_{\text{iter_count}}(i-1,j)$$
   $$i=2,3,\cdots,\text{W_ROWS};j=1,2,\cdots,(\text{W_COLS});$$

   5.4. 向右扩展栅格：

   $$\boldsymbol{W}_{\text{iter_count}}(i,j)=\boldsymbol{W}_{\text{iter_count}}(i,j)+\text{diffconst}*\boldsymbol{W}_{\text{iter_count}}(i,j+1)$$
   $$i=1,2,\cdots,\text{W_ROWS};j=1,2,\cdots,\text{W_COLS}-1;$$

   5.5. 向左扩展栅格：

   $$\boldsymbol{W}_{\text{iter_count}}(i,j)=\boldsymbol{W}_{\text{iter_count}}(i,j)+\text{diffconst}*\boldsymbol{W}_{\text{iter_count}}(i,j-1)$$

$i=1,2,\cdots,W_ROWS; j=2,3,\cdots,W_COLS;$

5.6. 若 $\boldsymbol{M}(i,j)=1$，则使得 $\boldsymbol{W}_{iter_count}(i,j)=0$；

$i=1,2,\cdots,W_ROWS; j=1,2,\cdots,W_COLS;$

5.7. 只要存在 $\boldsymbol{W}_{iter_count}(i,j)$ 首次大于高度，则设相应的 $\boldsymbol{H}_{iter_count}(i,j)=$ iter_count。

5.8. 将 $sum_count_{iter_count}$ 记为当前 $\boldsymbol{W}$ 矩阵中值大于 1 的所有元素之和。

5.9. IF $|sum_count_{iter_count}-sum_count_{(iter_count-1)}|<1$，THEN no_update_iter_count = no_update_iter_count + 1。

ENDIF

ENDWHILE

6. 所有值仍为零的 $\boldsymbol{H}(i,j)$ 点为障碍点。对这些点设置较大的值，即大于邻近点中的最大值（障碍点的下降梯度）。

7. 创建最短路径的坐标矢量 **sh_path_co ord_row** 和 **sh_path_co ord_col**，并初始化第一个点为 **sh_path_co ord_row**(1) = x_goal; **sh_path_co ord_col**(1) = y_goal。设置 μ。

8. 计算 $\boldsymbol{H}$ 矩阵在 x 方向上的梯度矩阵（$\nabla\boldsymbol{H}_x$）和 y 方向上的梯度矩阵（$\nabla\boldsymbol{H}_y$）。

9. $\nabla\boldsymbol{H}_x=-\nabla\boldsymbol{H}_x$；$\nabla\boldsymbol{H}_y=-\nabla\boldsymbol{H}_y$；path_index = 1；path_flag = 1。

10. WHILE(path_flag = 1)

10.1. 利用 $\nabla\boldsymbol{H}_y$ 矩阵通过插值来计算 del_row。

10.2. 利用 $\nabla\boldsymbol{H}_x$ 矩阵通过插值来计算 del_col。

10.3. 计算最短路径上下一点的坐标：

$$\boldsymbol{sh_path_co\ ord_row}(path_index+1)=\boldsymbol{sh_path_co\ ord_row}(path_index)+\mu^{*}\frac{del_row}{\sqrt{del_row^2+del_col^2}}$$

$$\boldsymbol{sh_path_co\ ord_col}(path_index+1)=\boldsymbol{sh_path_co\ ord_col}(path_index)+\mu^{*}\frac{del_row}{\sqrt{del_row^2+del_col^2}}$$

10.4. IF(达到初始点)THEN

path_flag = 0;

ENDIF

ENDWHILE

11. 对矢量 ***sh_path_co ord_row*** 和 ***sh_path_co ord_col*** 取反。

END

一旦最短路径确定后，需要寻找障碍物附近的角点。为确定这些角点，在该路径上取三个连续点，并计算前两点线段与后两点线段之间夹角的余弦。如果该余弦值小于给定阈值，则将这三个点的中间点作为角点，否则移动到下一个点，再次计算新角度的余弦值。不断执行该过程直到找到合适的角点。将该角点保存为导航的下一个子目标点。例如，在图 3.7 中，A1、A2 和 A3 是所考虑的三个点，则线段 $\overline{A1A2}$ 和 $\overline{A2A3}$ 夹角的余弦值非常大（大于给定阈值），因此 A2 不能看作一个角点。然后继续向前运动，当到达三个连续点 B1、B2 和 B3 时，线段 $\overline{B1B2}$ 和 $\overline{B2B3}$ 之间的角度足够大使得其余弦值小于给定阈值，则将 B2 看作一个角点。

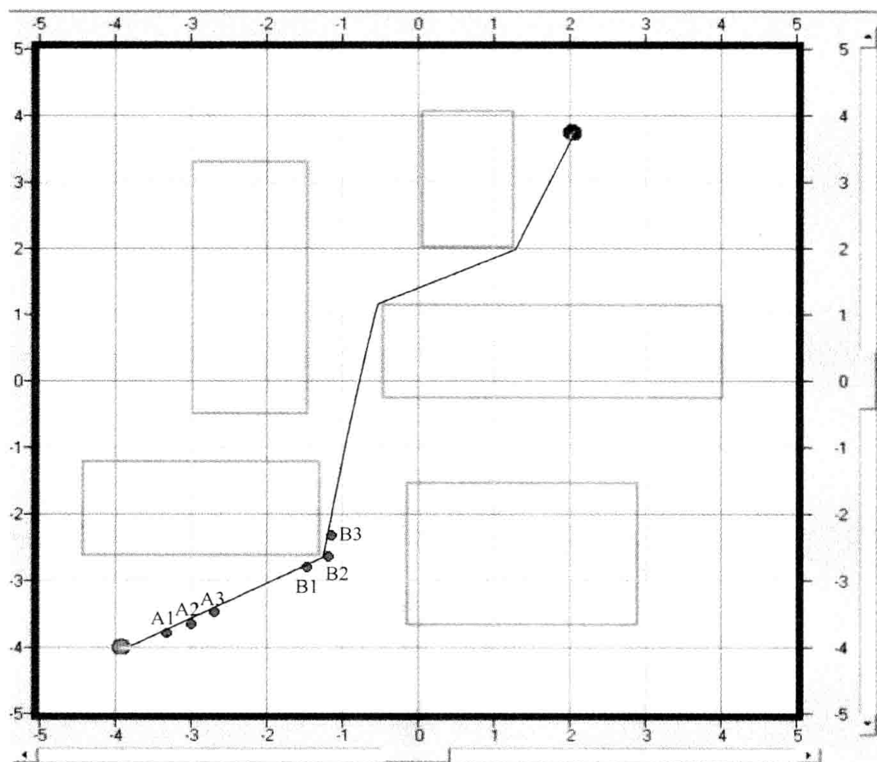


图 3.7　某一环境下最短路径计算示例

3.6　第二层中基于红外的导航

一旦子目标点确定后，控制就从第一层转移到第二层。接收到新的子目标信息后，机器人根据增量编码器信息来更新其当前位置（x_R，y_R，ϕ_R），并根据当前位置和子目标信息来确定新的转向角。理想情况下，这就是在静态场景中

机器人应旋转的角度，然后以恒定速度运动到子目标，这些子目标是在避障前提下由最短路径生成算法得到的一组点。然而，在动态场景中，激活基于视觉的地图构建子程序后，可能新的障碍已出现，或原先障碍物的位置已改变。这就可能导致在沿机器人和子目标之间的理想路径运动时会遇到障碍物。为解决动态环境下的问题，利用对称安装在 KOALA 机器人外围的十六个红外传感器进行导航导引。

这些红外传感器在机器人前方分布较密集，而在两侧分布较稀疏。图 3.8a

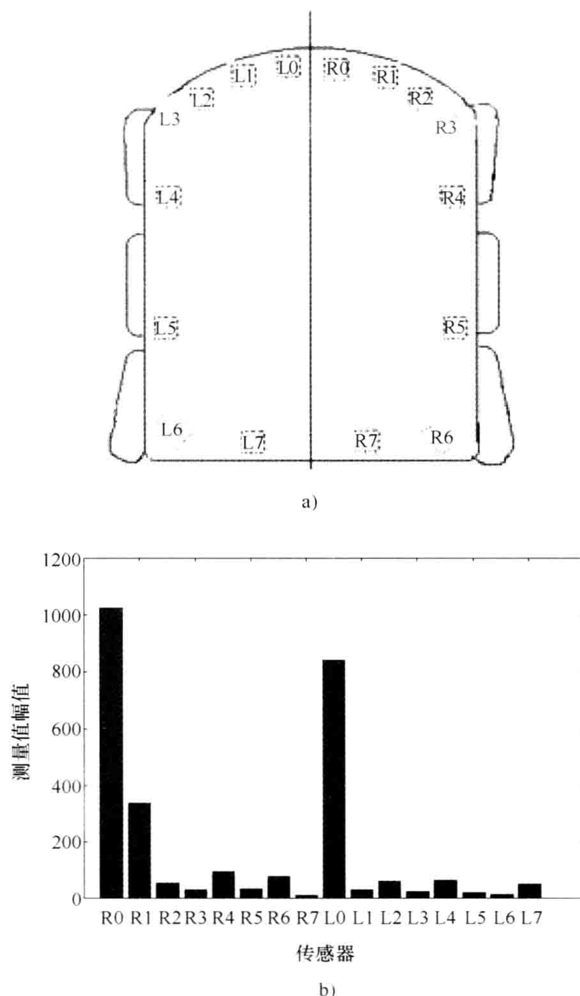


图 3.8 KOALA 机器人的红外传感器分布情况[23]
(见图 3.8a)以及在距离传感器(R0)前方 10cm 处放置一个
1.5cm 宽的障碍物时红外传感器的测量值(见图 3.8b)

给出了移动机器人的红外传感器分布情况，图3.8b给出了在机器人前方10cm处放置一个1.5cm宽的障碍物时红外传感器测量值的典型情况。这十六个红外传感器在导航时对环境进行扫描。根据这些传感器读数，系统计算机器人前方的障碍区域和自由区域，并由这些计算值来确定机器人的可运行区域。为确定可运行区域，对这十六个传感器每个都分别设置各自的阈值，并且设传感器(R0～R3，L0～L3)的优先级最大。对于每个红外传感器，如果读数大于其阈值，则意味着前方有障碍，否则认为是可运行的自由区域。现在假设，根据这些读数可知在机器人当前位置的左侧和右侧都是可运行区域，这时就应根据机器人相对其理想运行方向转向最小的方向来决定机器人应左转还是右转。一旦确定绕行方向，就可根据该方向上的红外传感器读数来确定机器人的运动速度。机器人运行了一段预先设定的距离后，重新激活红外扫描，并重新确定运行的绕行方向，不断执行上述过程直到机器人到达子目标或其最邻近点，然后机器人停止运行，并重新返回到第一层的控制。

3.7　实际性能评估

在实验室环境下，利用几种场景对基于视觉的导航进行性能评估。在此，给出四种环境下的实验结果，其中在静态环境下进行两次实验，另外两次实验是在动态环境下完成的。

1. 实验分析Ⅰ

机器人的初始位置为(0，0，0)，目标点为(2，0)。在机器人和目标位置之间存在一个障碍物。在此需要注意的是，对于配置有一个云台系统以及相应自由度的机器人系统，应在某一特定环境下合理地初始化水平角和俯仰角，并在随后的所有实验中保持不变。初始时，对所开发的系统选择这两个角度使得在单目摄像头的每帧图像中都能覆盖相当大的地面和环境区域。因此，本系统具有一定的灵活性，可根据导航环境合理地初始化这些角。图3.9a给出了从摄像头视频流中采集的图像帧，图3.9b～f分别给出了机器人在初始位置时的图像处理步骤的顺序。机器人在其初始位置时，由机器人所观察的地面上障碍物的边缘实际上是从(0.9，0.3)到(0.9，-0.85)。图3.10给出了具有障碍物信息和自由区域的栅格地图。根据该地图来计算最短路径，并用子目标信息更新机器人导航算法的第二层。在此，根据算法计算的子目标1为(0.819，0.33)。在控制转移到第二层后，就利用红外传感器将机器人导航到子目标1。这时，由于与计算的子目标有很小的偏差，机器人实际停止在(0.819，0.332)处。图3.11给出了机器人达到第一个子目标点时的栅格地图。在控制又返回到第一层时构建该栅格地图，并再次执行基于视觉的处理过程。图3.12a～d给出了机器人在第

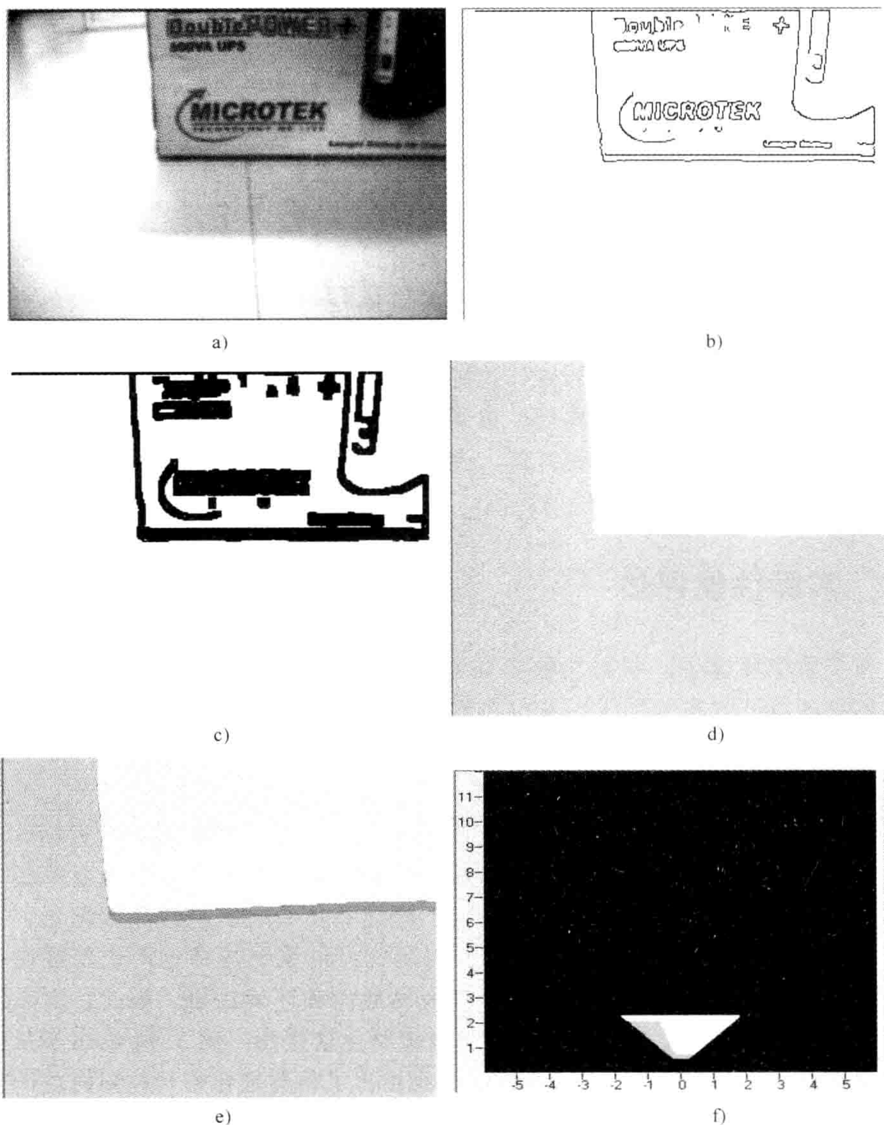


图 3.9 采集的图像(见图 3.9a)和在第一层执行的图像处理序列[边缘图像(见图 3.9b)，边缘加粗的图像(见图 3.9c)，区域增长图像(见图 3.9d)，包括自由区域、障碍区域和隐藏区域的图像(见图 3.9e)以及梯形地面图像(见图 3.9f)]

一个子目标点处执行图像处理步骤后的结果，这些结果又用于基于红外传感器的导航，不断重复上述过程直到到达最终的目标点。图 3.13 给出了机器人到达最终目的地时的栅格地图。机器人最终停止在(1.963, 0.024)处，非常接近于

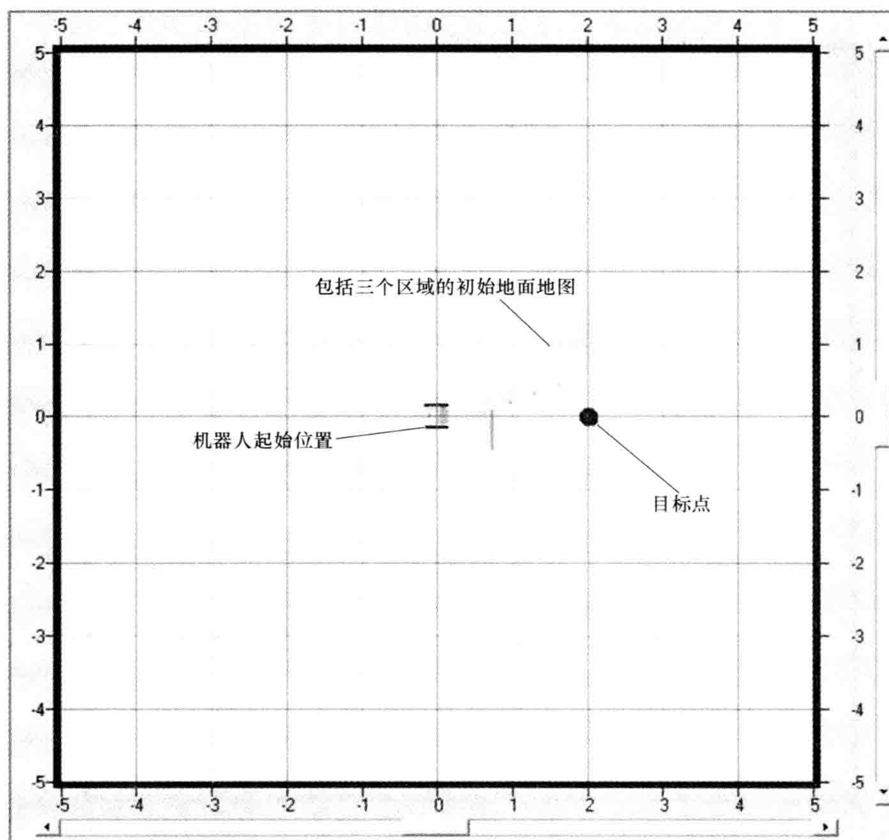


图 3.10　初始时的栅格地图

指定的目标点(2，0)。图 3.14 给出了机器人从起始点开始在存在障碍物情况下沿最短路径到达目标点所实际经过的完整导航路径。图 3.15a 给出了导航过程中机器人右侧红外传感器(R0，R3)的响应曲线，图 3.15b 给出了机器人左侧红外传感器(L0，L3)的响应曲线。由图可知，当机器人在障碍物附近时，R3 的读数较高。随着机器人绕过障碍物向目标点运动时，R3 读数又逐渐减小。

2. 实验分析Ⅱ

在本实验中，机器人的初始位置同样为(0，0，0)，新的目标点为(3，0)。在机器人和目标位置之间放置两个障碍物。图 3.16a 给出了机器人在起始位置时采集的一帧图像，图 3.16b ~ f 给出了在第一层执行图像处理步骤的结果。图 3.17 给出了具有障碍信息和自由区域的初始栅格地图。接下来，计算最短路径，并根据该子目标信息实现机器人导航算法的第二层。图 3.18 给出了机器人到达第一个子目标点时的栅格地图。在机器人到达子目标 1 后，控制返回到第一层。这时，系统再次执行基于视觉的处理过程，如图 3.19 所示。图 3.20 给出了机器

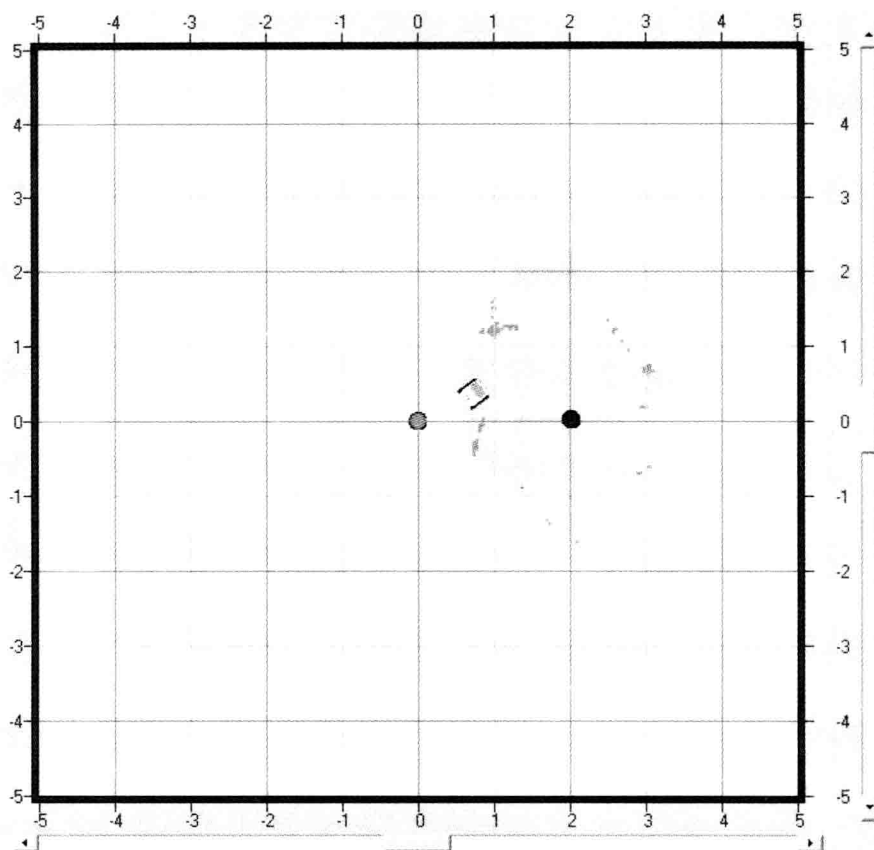


图 3.11　机器人到达子目标 1 时的栅格地图

人利用第二层中的红外导航到达子目标 2 时的栅格地图。不断重复执行该迭代过程直到机器人到达最终的目标点。图 3.21 给出了机器人到达最终目标点时的栅格地图。图 3.22 给出了机器人在该静态环境中所经过的完整路径，表明机器人能够圆满地到达目标点。图 3.23 给出了在机器人朝目标点的导航过程中，四个红外传感器 R0、R3、L0 和 L3 读数的变化曲线。

3. 实验分析Ⅲ

在接下来的两个实验中，展示了本书所开发的系统在动态变化环境中的应用。在此，在一个类似于实验Ⅰ的环境中，机器人从起始位置(0，0，0)开始，在机器人和目标位置之间存在障碍物的情况下，试图到达目标点(2，0)。然而，在机器人开始基于红外导航朝子目标 1 运动后，通过第一层中基于视觉的图像处理来确定机器人起始位置，并利用最短路径算法确定子目标 1 时，障碍物 1 的位置发生了变化。新的障碍物位置如图 3.24 所示，这时该障碍物运动到机器人附近，并相对于其初始位置转向机器人的左侧。由于环境中的动态变化，机器

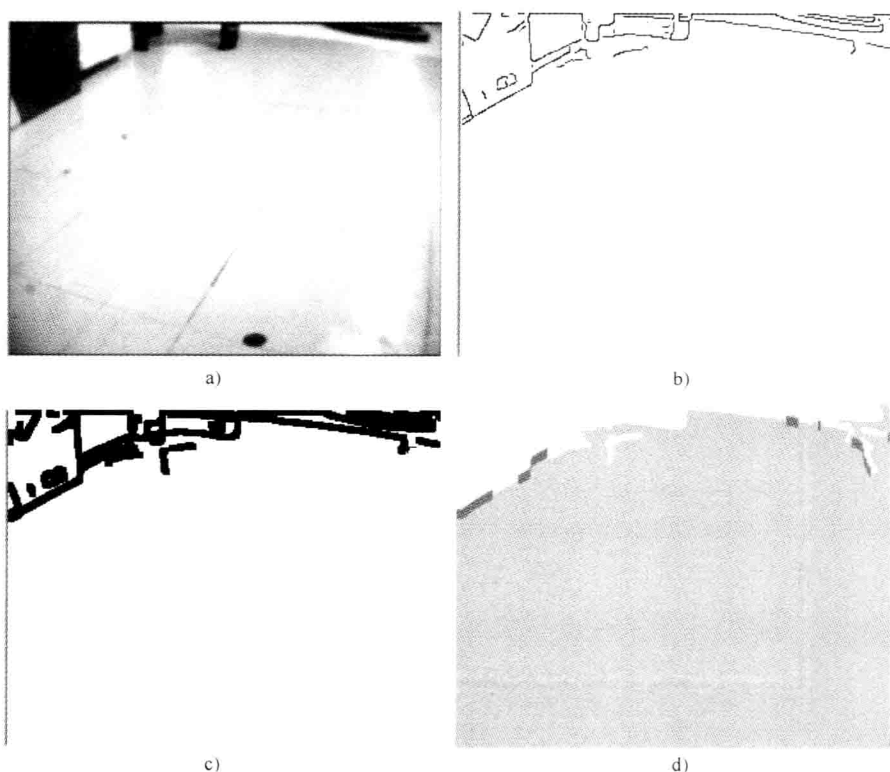


图3.12　在子目标1处的图像处理结果

a）采集的图像　b）边缘图像　c）边缘加粗的图像　d）包含三个不同区域的图像

人向其左侧转向，但仍能够避开障碍并到达子目标。随后再次激活迭代算法，结果表明机器人能够几乎完美地到达其最终目标。图3.24给出了机器人在动态环境中的导航。图3.25给出了位于机器人前方的红外传感器读数。由图可看出，在动态环境中有障碍物运动时，R0和L0的读数会产生突变。

在此，应注意的是，如果存在动态变化的障碍物正好运动到子目标处的特殊情况时，则根据该算法，实际的红外导航机制将能够保证机器人在满足避障或避碰的前提下停止在距离子目标最近的位置。

4. 实验分析Ⅳ

本实验的情况类似于实验Ⅱ，只是存在有自然运动的障碍物。在此，机器人开始朝子目标1运动后，在避开由第一层基于视觉的图像处理确定的障碍物1时，障碍物1的位置突然变化。该障碍物向机器人左侧运动并接近机器人，这时在自由区域中会产生动态变化的区域。同理，在机器人沿最短路径运行并避开障碍物2时，障碍物2的位置会突然变化，使得更接近机器人。然而，机器人在每种情况下都能够实现正确转向，并圆满地到达最终目标，如图3.26所示。

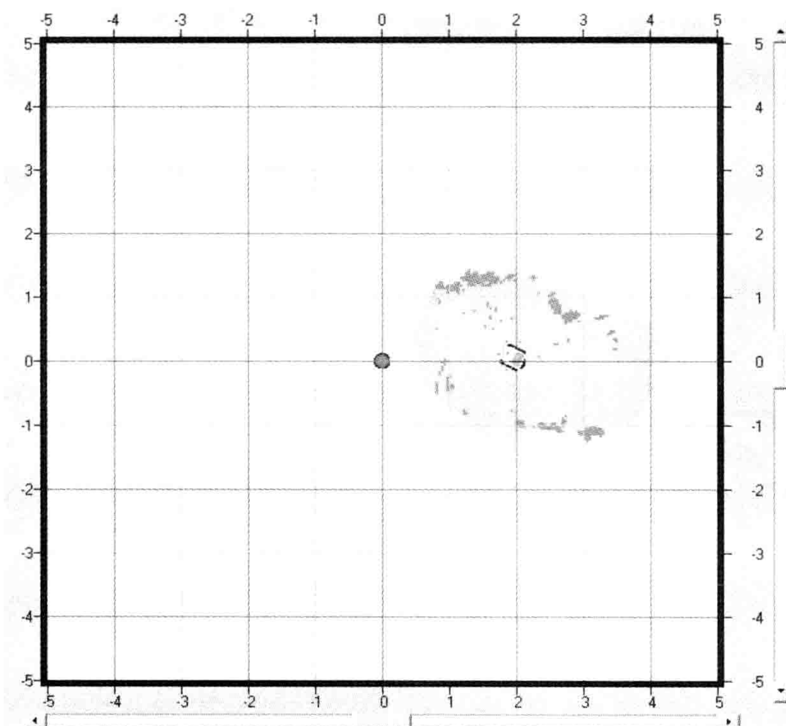


图 3.13　机器人到达最终目标点时的栅格地图

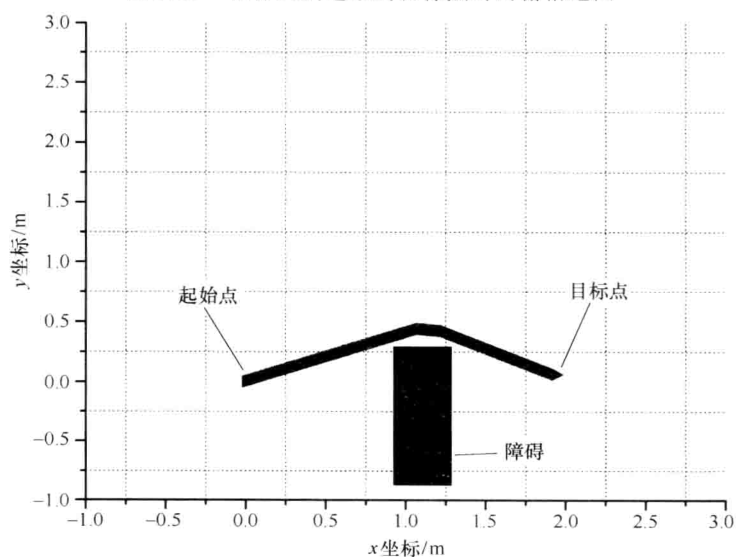


图 3.14　机器人所经过的导航路径

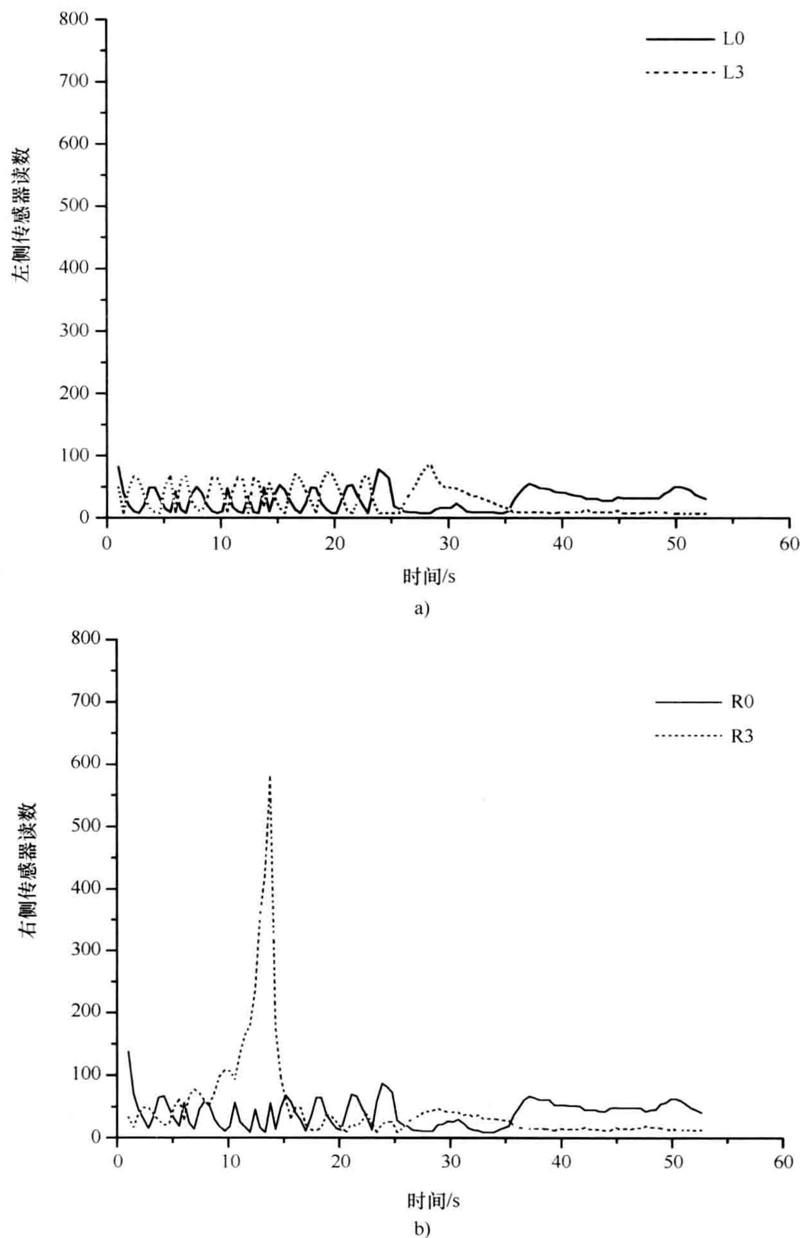


图 3.15　实验 I 中红外传感器 L0 和 L3 随时间变化的响应曲线
(见图 3.15a)以及红外传感器 R0 和 R3 随时间变化的响应曲线(见图 3.15b)

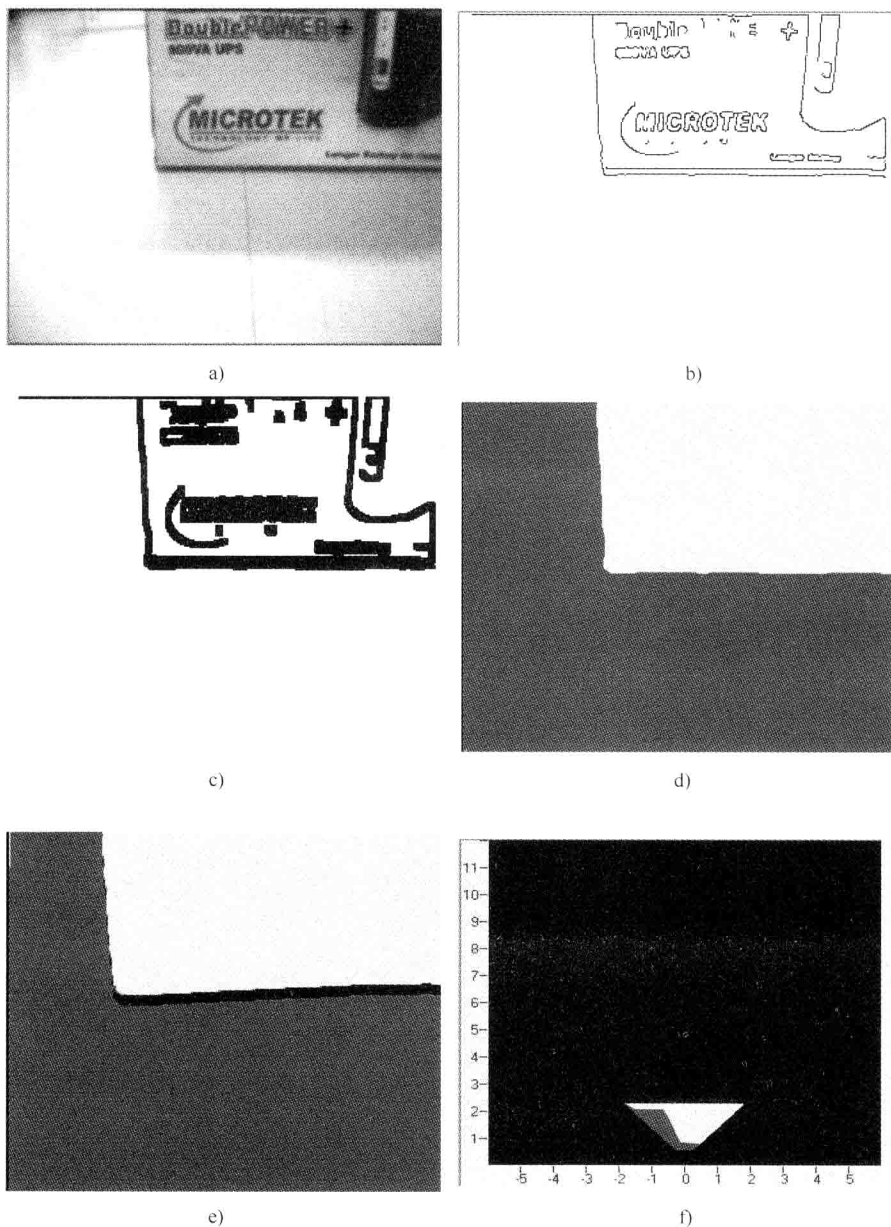


图 3.16　采集的图像(见图 3.16a)和在第一层中执行图像处理步骤的结果［边缘图像(见图 3.16b)，边缘加粗的图像(见图 3.16c)，区域增长图像(见图 3.16d)，包含自由区域、障碍区域和隐藏区域的图像(见图 3.16e)以及梯形地面图像(见图 3.16f)］

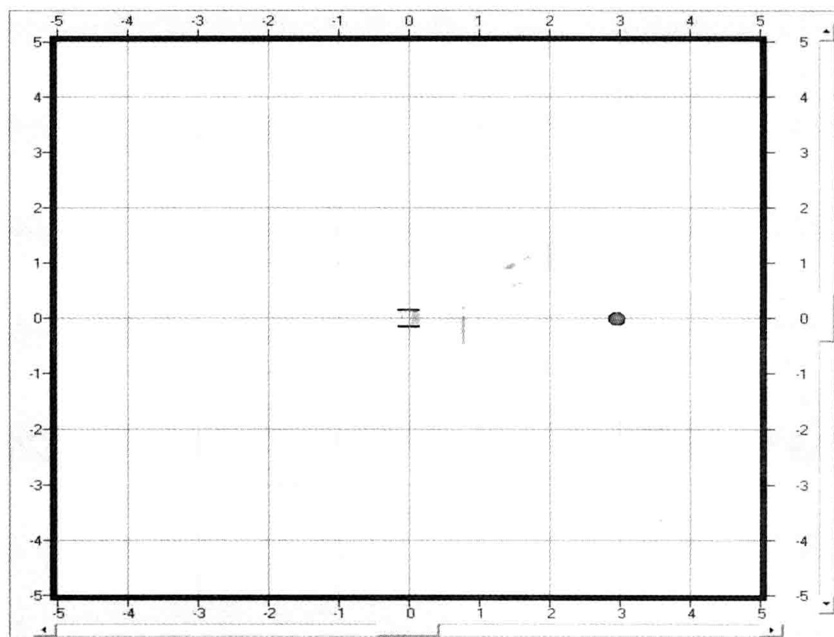


图 3.17　初始时的栅格地图

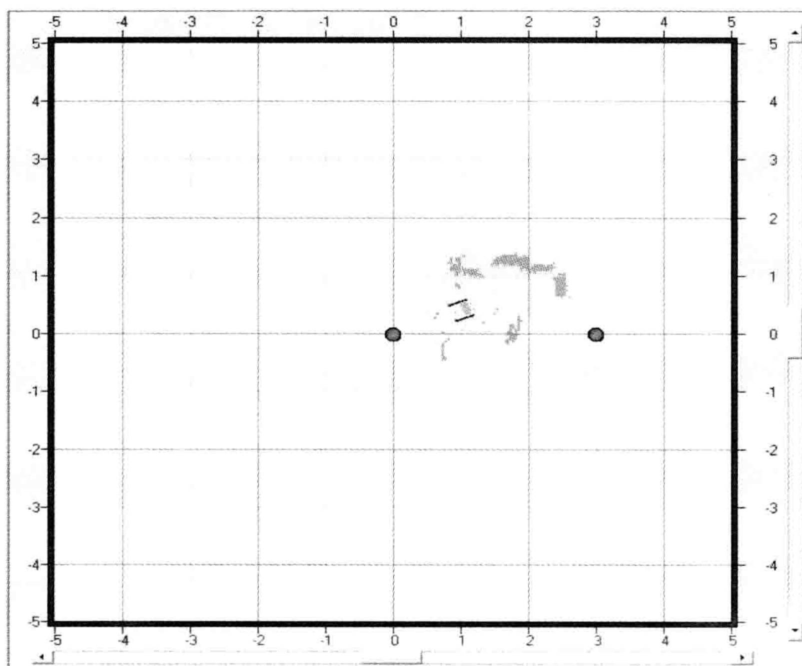


图 3.18　机器人到达子目标 1 时的栅格地图

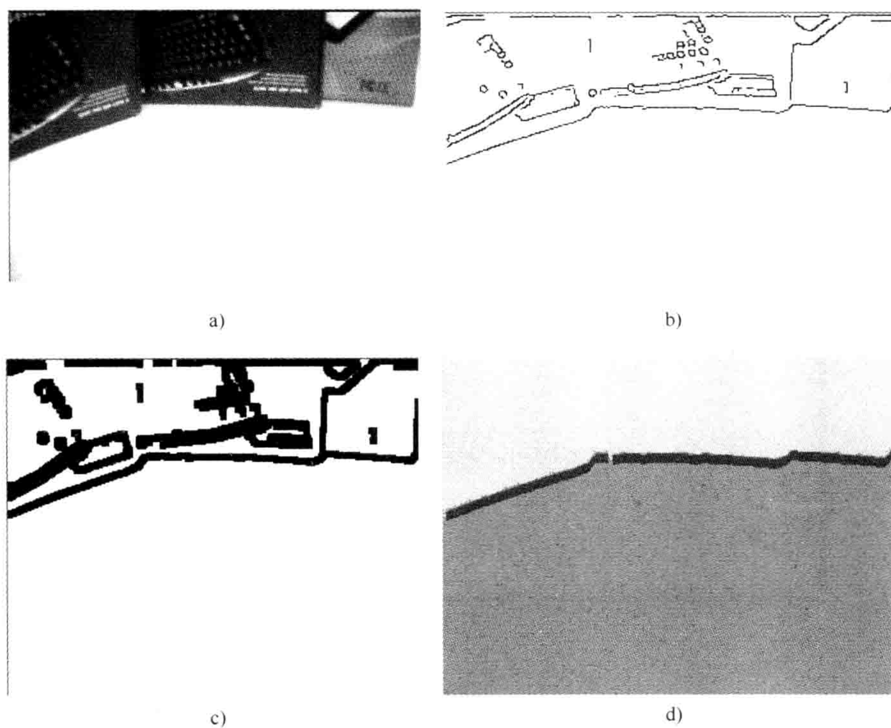


图 3.19 在子目标 1 处的图像处理结果

a)采集的图像 b)边缘图像 c)边缘加粗的图像 d)包含三个不同区域的图像

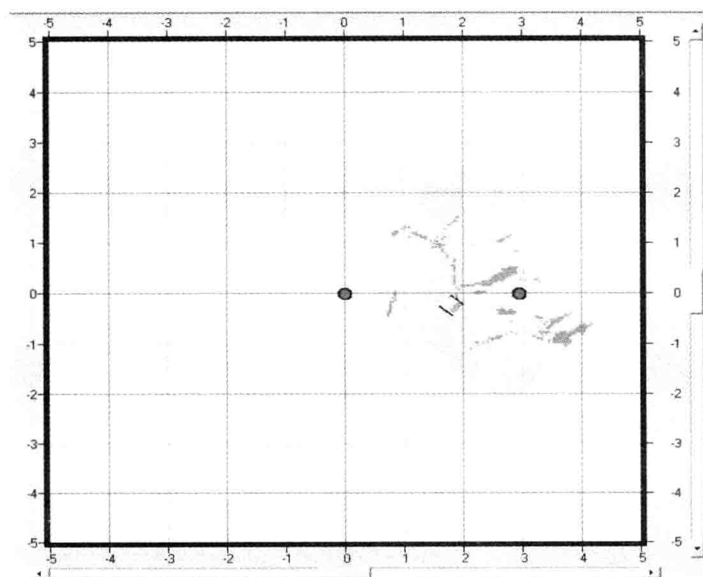


图 3.20 机器人到达子目标 2 时的栅格地图

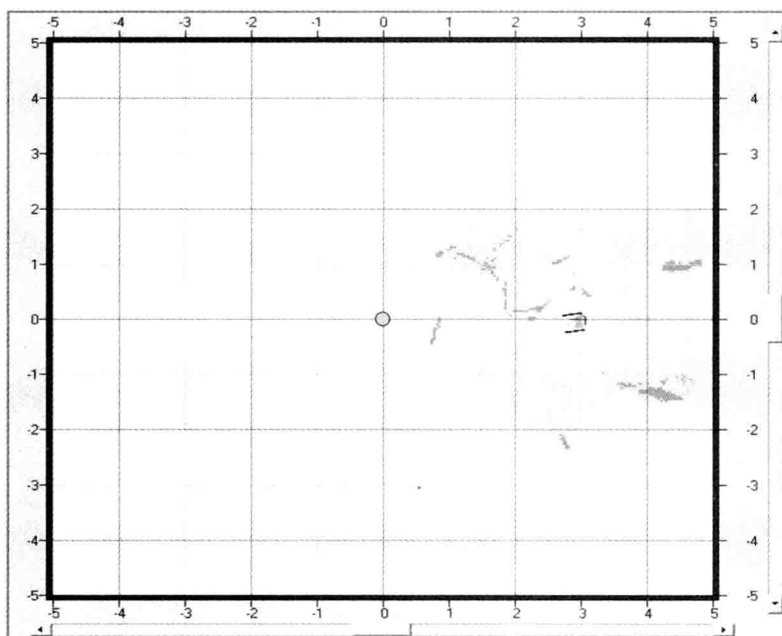


图 3.21　机器人达到最终目标点时的栅格地图

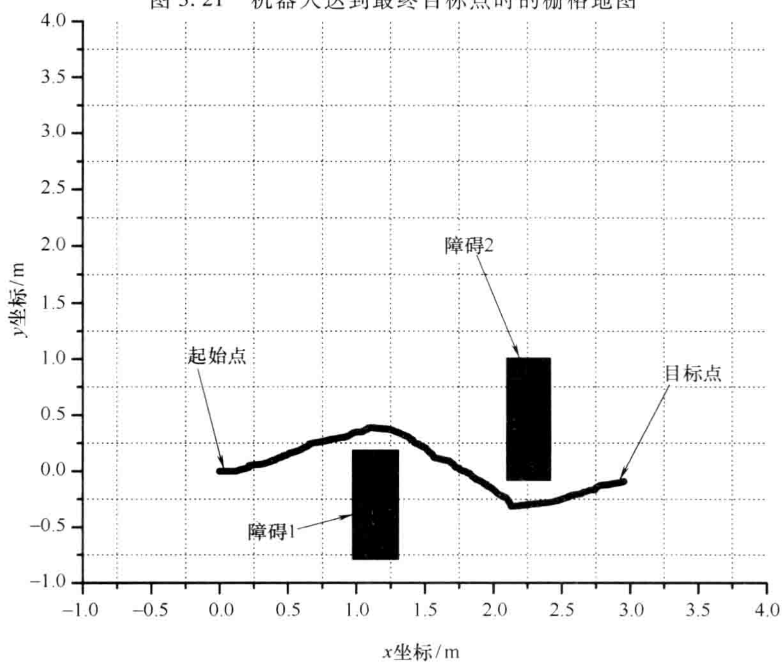
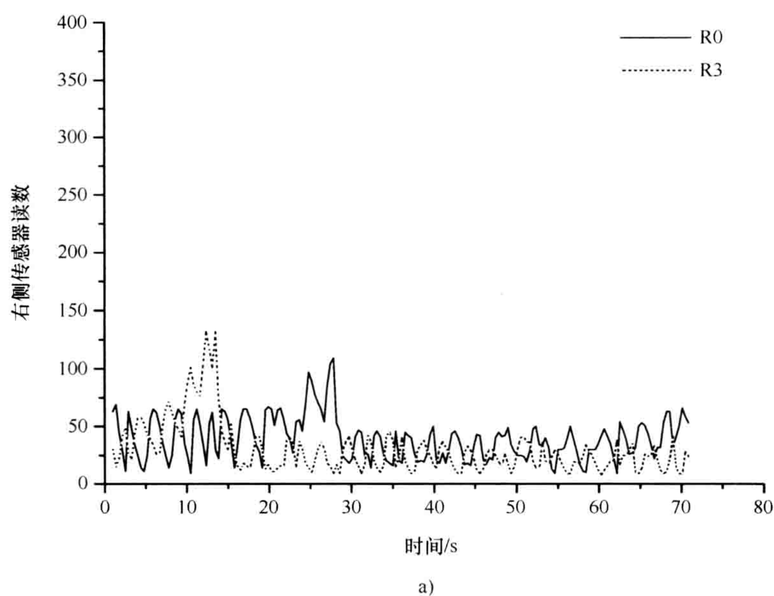
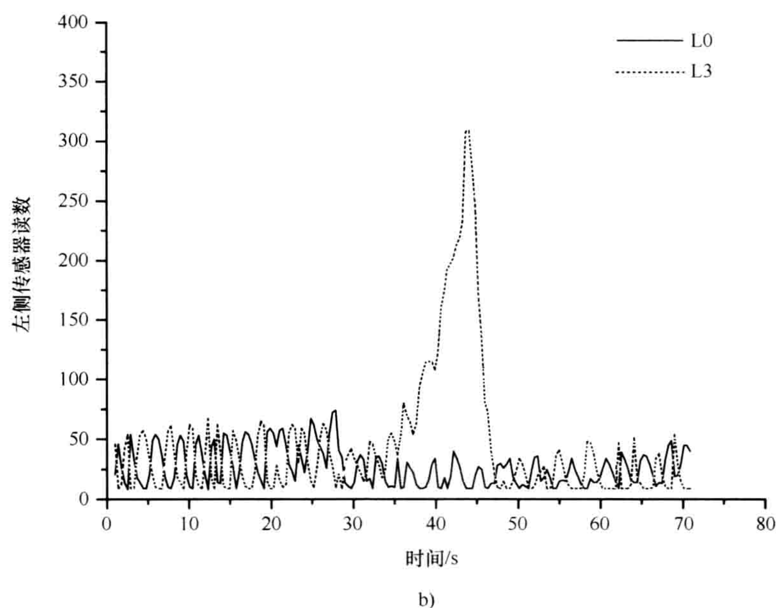


图 3.22　实验 Ⅰ㊀中机器人所经过的导航路径

㊀ 此处原书有误，应为实验Ⅱ。——译者注



a)



b)

图 3. 23 在实验Ⅱ的导航过程中红外传感器 R0 和 R3 随时间变化的响应曲线(见图 3. 23a)以及红外传感器 L0 和 L3 随时间变化的响应曲线(见图 3. 23b)

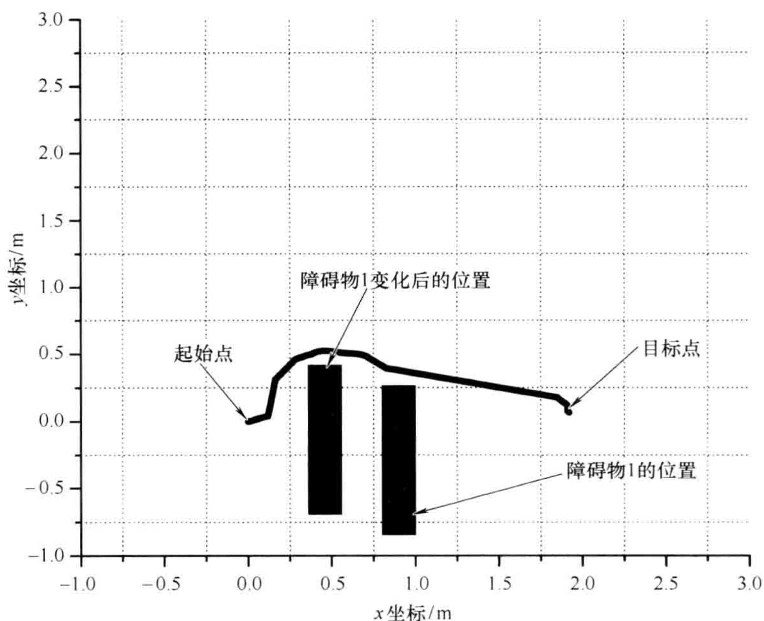


图 3.24　动态环境中机器人所经过的导航路径

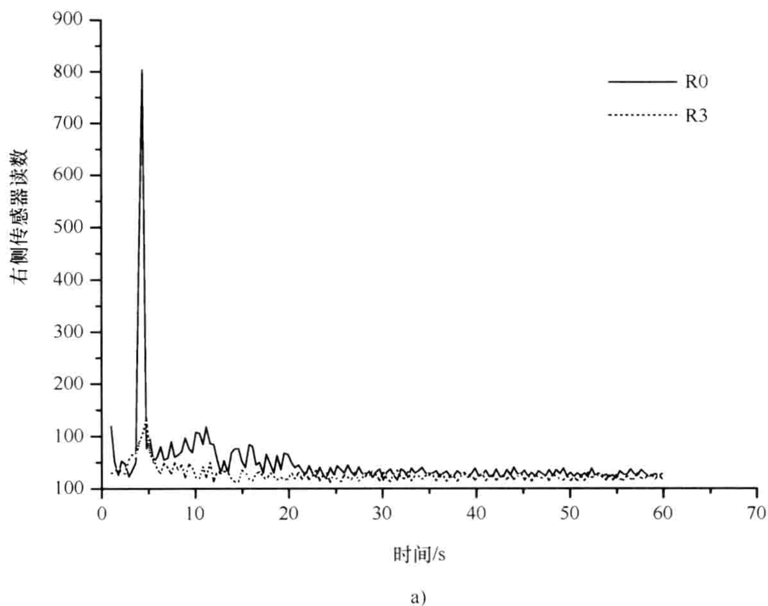
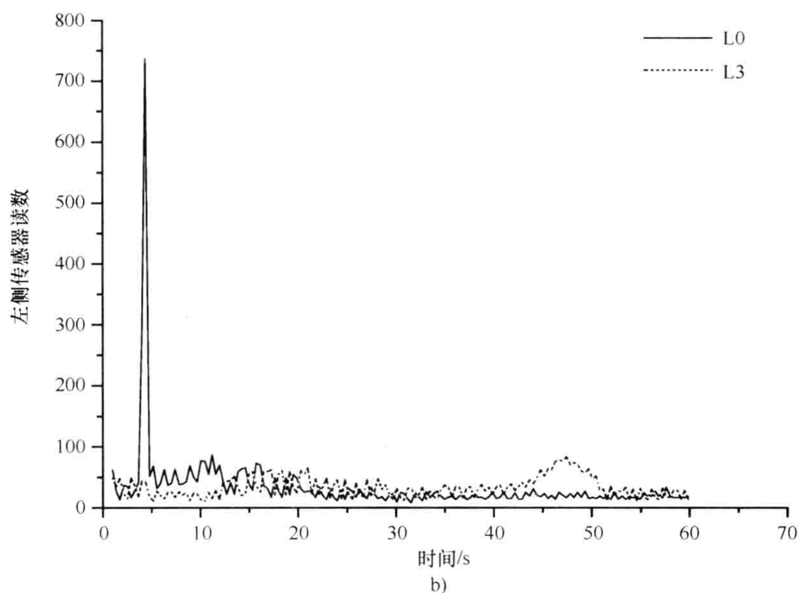


图 3.25　在实验Ⅲ的动态环境导航过程中红外传感器 R0 和 R3 随时间变化的响应曲线(见图 3.25a)以及红外传感器 L0 和 L3 随时间变化的响应曲线(见图 3.25b)



b)

图 3.25 在实验Ⅲ的动态环境导航过程中红外传感器R0和R3随时间变化的响应曲线(见图3.25a)以及红外传感器L0和L3随时间变化的响应曲线(见图3.25b)(续)

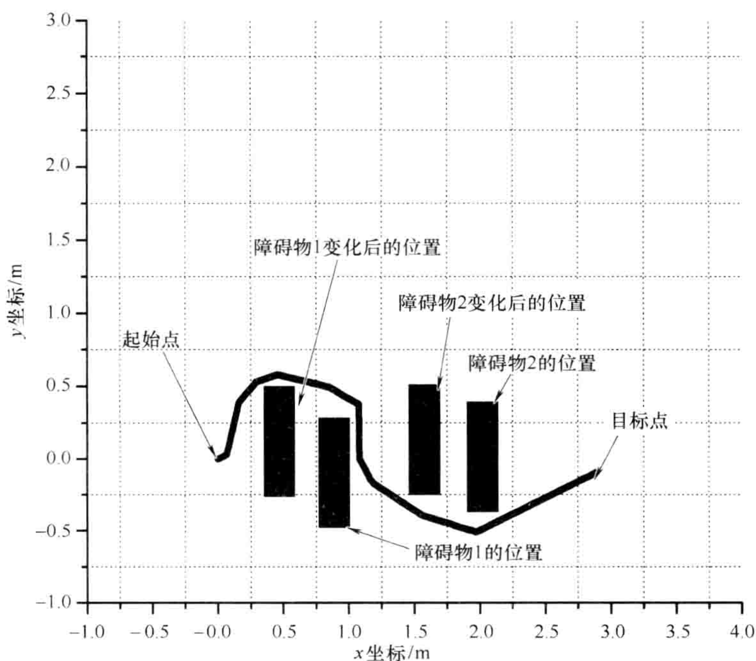


图 3.26 动态环境中机器人所经过的导航路径

图3.27给出了红外传感器R0、R3、L0和L3的读数。由图可以看出，当两个障碍物的位置发生变化时，L0和R0的读数也产生两个突变。

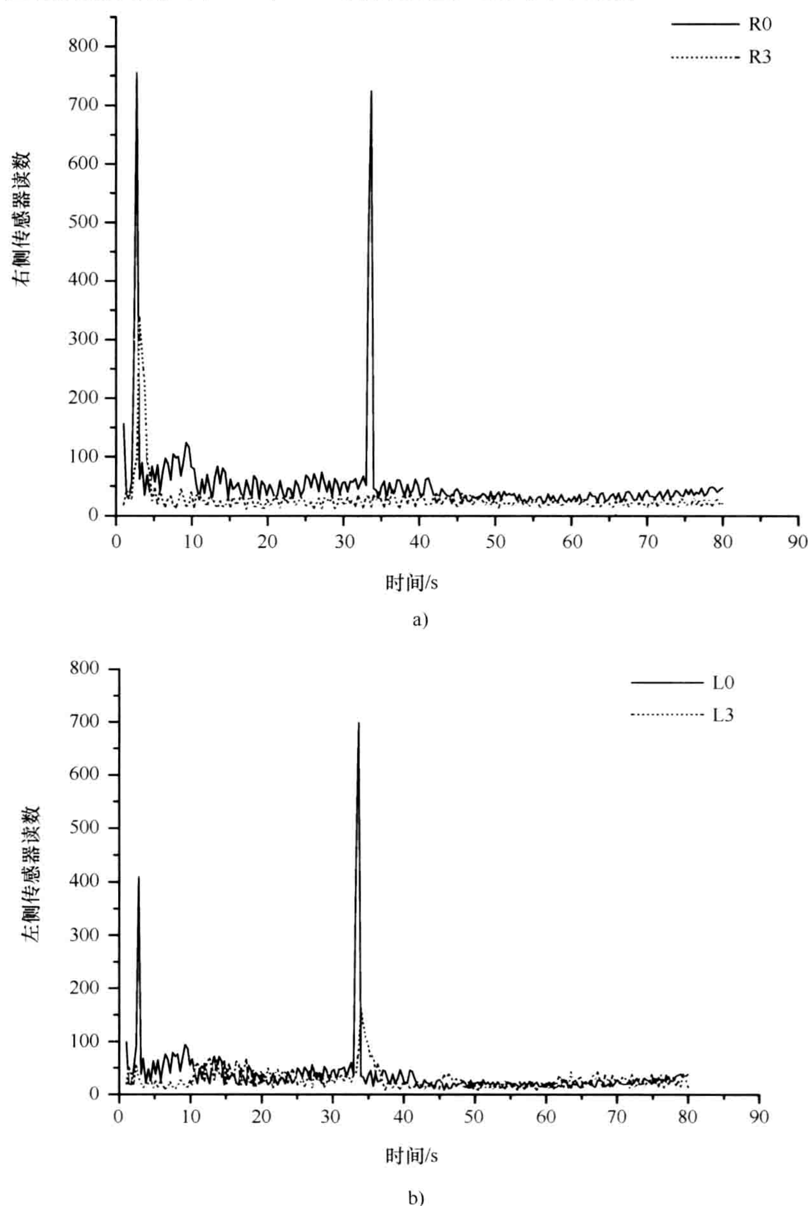


图3.27 在实验Ⅳ的动态环境导航过程中红外传感器R0和R3随时间变化的响应曲线(见图3.27a)以及红外传感器L0和L3随时间变化的响应曲线(见图3.27b)

3.8 本章小结

本章介绍了如何开发一个目标导向的基于视觉的两层导航机制。该系统在第一层中采用基于视觉的环境分析，其中利用几种图像处理函数以及一个最短路径生成算法来确定导航的下一个子目标，并在避障前提下到达最终目标。在第二层中利用该子目标信息在动态环境中进行机器人导航，其中利用一组红外传感器来实现避障并到达子目标或其邻近点。迭代执行该两层算法来创建下一个子目标，并将机器人导航到该目标，从而实现快速到达最终目标。本章给出了一种如何综合利用基于摄像头的图像处理方法和最短路径生成算法来提高机器人在实际环境中视觉导航效果的成功实现，从而使得机器人沿着实际最短路径在避障前提下到达其最终目标(已知其先验知识)。系统的鲁棒性可由基于红外的导航进一步保证，这能有助于机器人在障碍物位置改变的动态环境下自适应导航。最后，在实验室的几种室内环境下应用了该算法。实验结果表明 KOALA 机器人可在静态环境和动态环境下圆满地完成任务。

所开发的程序包括了 VB 平台下开发的能够与机器人系统的处理器(下载了用户编程的交叉编译 C 程序)实时通信的高层程序。然而，在实际的实现阶段，整个系统可通过一个人机友好的 GUI 程序在 PC 的 VB 平台上运行，从而普通用户能够很容易地应用该系统。

对于光照较好的情况，正如所期望的，算法能够提供比较满意的性能。然而，对于光照条件较差的情况，地面上障碍物的反射可能较暗(见图 3.4a ~ e)，从而使得边缘图像可能对应地面反射的边缘。因此，这些反射可看作障碍物，从而使得所计算的自由区域缩小。然而根据算法，在这些情况下，计算得到的最短路径可能比实际最短路径要长，但仍能够保证在避障前提下机器人导航的安全性和鲁棒性。

本系统是在真实环境下的机器人位置仅通过增量编码信息由里程计估计而得的情况下开发的，这在地面均匀的室内环境下的应用效果很好。所进行的实验充分表明在上述条件下，对于不同的环境配置，机器人可以到达实际环境中的目标。然而，由于车轮打滑等问题，在室外环境中该系统的准确性可能会受到影响。在接下来的研究工作中应考虑如何使得该系统同样适用于室外环境，这可能会通过在当前系统中集成用于机器人自定位的 EKF 算法来实现。

致谢 本章所介绍的工作得到了印度大学教育资助委员会重大研究项目计划的资助[资助号：32-118/2006(SR)]。

参考文献

[1] Chen, Z., Birchfield, S.T.: Qualitative vision-based mobile robot navigation. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), Orlando, Florida (May 2006)

[2] Bertozzi, M., Broggi, A., Fascioli, A.: Vision-based intelligent vehicles: state of the art and perspectives. Robotics and Autonomous Systems 32, 1–16 (2000)

[3] Shin, D.H., Singh, S.: Path generation for robot vehicles using composite clothoid segments. The Robotics Institute, Internal Report CMU-RI-TR-90-31, Carnegie-Mellon University (1990)

[4] DeSouza, G.N., Kak, A.C.: Vision for mobile robot navigation: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(2), 237–267 (2002)

[5] Moravec, H.P.: Sensor fusion in certainty grids for mobile robots. Artificial Inteligence Magazine 9, 61–74 (1988)

[6] Elfes, A.: Sonar-based real-world mapping and navigation. IEEE Journal of Robotics and Automation 3(6), 249–265 (1987)

[7] Jensfelt, P., Christensen, H.I.: Pose tracking using laser scanning and minimalistic environmental models. IEEE Transactions on Robotics and Automation 17, 138–147 (2001)

[8] Yamauchi, B., Beer, R.: Spatial Learning for navigation in dynamic environments. IEEE Transactions on System, Man, and Cybernetics, Part B 26(3), 634–648 (1995)

[9] Pierce, D., Kuipers, B.: Learning to explore and build maps. In: Proc. of the Twelfth National Conference on Artificial Intelligence, Menlo Park, pp. 1264–1271. AAAI, AAAI Press/MIT Press (July 1994)

[10] Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence 99, 21–71 (1998)

[11] Santos-Victor, J., Sandini, G., Curotto, F., Garibaldi, S.: Divergent stereo for robot navigation: Learning from Bees. In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, pp. 434–439 (June 1993)

[12] Kim, D., Nevatia, R.: Recognition and localization of generic objects for indoor navigation using functionality. Image and Vision Computing 16(11), 729–743 (1998)

[13] Murray, D., Little, J.J.: Using real-time stereo vision for mobile robot navigation. Autonomous Robots 8, 161–171 (2000)

[14] Davison, A.J.: Mobile robot navigation using active vision. PhD thesis (1998)

[15] Ayache, N., Faugeras, O.D.: Maintaining representations of the environment of a mobile robot. IEEE Transactions on Robotics and Automation 5(6), 804–819 (1989)

[16] Fialaa, M., Basub, A.: Robot navigation using panoramic tracking. Pattern Recognition 37, 2195–2215 (2004)

[17] Gasper, J., Santos- Victor, J.: Vision-based navigation and environmental representations with an omnidirectional camera. IEEE Transactions on Robotics and Automation 16(6), 890–898 (2000)

[18] Ohya, A., Kosaka, A., Kak, A.: Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. IEEE Transactions on Robotics and Automation 14(6), 969–978 (1998)

[19] Li, M.H., Hong, B.H., Cai, Z.S., Piao, S.H., Huang, Q.C.: Novel indoor mobile robot navigation using monocular vision. Engineering Applications of Artificial Intelligence, 1–18 (2007)
[20] Latombe, J.: Robot Motion Planning. Kulwer, Norwell (1991)
[21] Dijkstra, E.W.: A note on two problems in connection with graphs. Numerische Mathematik 1, 269–271 (1959)
[22] Nilsson, N.J.: Principles of Artificial Intelligence. Tioga Publishing Company (1980)
[23] KOALA User Manual, Version 2.0(silver edition), K-team S.A., Switzerland (2001)
[24] Singh, N.N., Chatterjee, A., Rakshit, A.: A PIC microcontroller-based system for real-life interfacing of external peripherals with a mobile robot. International Journal of Electronics 97(2), 139–161 (2010)
[25] Kim, P.G., Park, C.G., Jong, Y.H., Yun, J.H., Mo, E.J., Kim, C.S., Jie, M.S., Hwang, S.C., Lee, K.W.: Obstacle Avoidance of a Mobile Robot Using Vision System and Ultrasonic Sensor. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. LNCS, vol. 4681, pp. 545–553. Springer, Heidelberg (2007), doi:10.1007/978-3-540-74171-8.
[26] http://www.mathworks.com/matlabcentral/fileexchange/8625-shortest-path-with-obstacle-avoidance-ver-1-3
[27] http://atalasoft-imgx-controls-sdk.software.informer.com/
[28] Nirmal Singh, N., Chatterjee, A., Chatterjee, A., Rakshit, A.: A two-layered subgoal based mobile robot navigation algorithm with vision system and IR sensors. Measurement 44(4), 620–641 (2011)
[29] Nirmal Singh, N.: Vision Based Autonomous Navigation of Mobile Robots. Ph.D. Thesis, Jadavpur University, Kolkata, India (2010)

第 4 章　基于视觉的移动机器人自主开发

摘要：本章将介绍如何在实验室环境下自行开发一种具有特殊功能的低成本机器人。尤其是将详细介绍两种能与机器人集成的基于 PIC 微控制器的传感器系统的开发过程：其中一种是具有动态测距增强功能的红外测距系统；另一种是应用开关模式同步检测技术的光学接近检测系统。

4.1　简介

在 KOALA 机器人（实验室购置的 KOALA 机器人）上实现基于视觉的导航算法过程中，发现对 KOALA 机器人的操作具有一定约束条件，具体如下：

1）通过 RS232 实现 PC/笔记本电脑与 KOALA 机器人之间的通信，然而现今大多数的 PC/笔记本电脑都已不再配置串口，因此需要一个独立的 USB—串口转换器来实现与 KOALA 机器人之间的操作。

2）在 PC 与 KOALA 机器人之间无法高速传输数据。

3）KOALA 机器人的 I/O 接口有限。

4）KOALA 机器人没有提供任何 USB 接口。

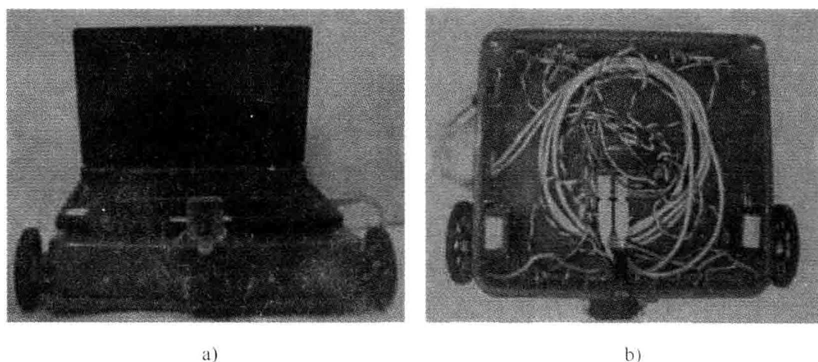
5）低成本的 USB 网络摄像头无法与 KOALA 机器人直接相连。

6）KOALA 机器人配置的低端处理器不能进行图像处理，因此需要一个具有无线摄像头接口的车载笔记本电脑或 PC，这样会使得结构复杂且体积庞大。

因此，本实验室旨在自主研制开发一种能够克服上述缺点的机器人，其具体功能和能力将在本章详细介绍。

4.2　基于视觉的低成本移动机器人的研制开发

如上所述，为了对工业行业提供一种低成本的解决方案，本实验室自行研制了一种移动机器人[14]。图 4.1 给出了实际机器人的前视图和底视图。机器人的结构框图如图 4.2 所示。所研制开发的机器人是一个两轮的差动驱动系统，该机器人配置了六个红外接近传感器、一个红外测距系统和一台笔记本电脑。接近传感器提供了一种布尔信号，若机器人与障碍物之间的距离足够近，则每个传感器均被激活，反之则保持禁用状态。由于红外传感器系统的角位置由一



a) b)

图 4.1 自主开发的移动机器人
a)前视图 b)底视图

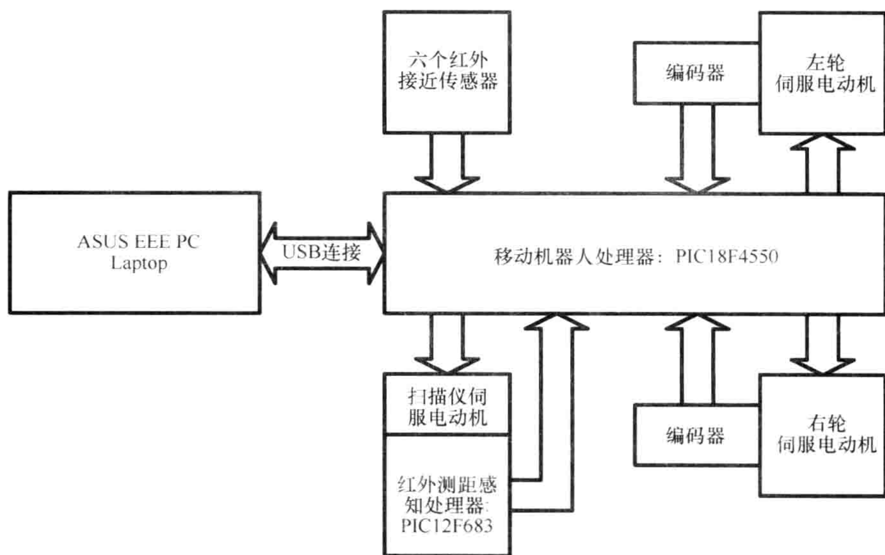


图 4.2 移动机器人的原理示意图

个伺服电动机控制，由此对整个系统增加了一个自由度，这样就能使得红外传感器在十一个角位置从左到右地扫描机器人前方和两侧的环境。在机器人系统中配置一台笔记本电脑，从而使其成为一个独立自主的系统。该笔记本电脑包括4GB固态硬盘、1GB内存和Windows XP SP2操作系统，且与任何运动部件均独立，并通过USB与机器人本体进行通信。机器人本体由笔记本电脑通过两根USB线供电(5V，1A)，因此移动机器人的运行无需独立电源。对所有RC伺服电动机均能通过电能控制来节能，利用霍尔效应开关来实现左、右轮的编码器(4脉冲/r)。利用一个具有自动对焦的笔记本电脑摄像头可作为机器人系统的单目视觉传感器。另外，还专门开发了用于障碍物检测和避障的红外测距感知系

统，该系统采用一个自主开发的微控制器系统(PIC12F683)来提高测距系统的动态范围。同时，该系统还采用一个基于 Visual Basic 的机器人控制程序并结合视觉和红外测距传感器来实现机器人导航功能。另外，该系统还配置了 WiFi 用于无线远程监测与监控。

4.3　基于微控制器的机器人感知系统的研制开发

本书所开发的机器人具有三种特殊功能。首先，该机器人包括自主研发的两种特殊类型的传感器系统：具有动态测距增强功能的红外(IR)传感器[2]；采用开关模式同步检测技术的光学接近检测器[15]。这两种传感器系统都是基于 PIC 微控制器开发的。此外，机器人系统还具有内网连接的复杂功能，其中工作在从模式下的机器人上配置的笔记本电脑可由工作在主模式下的远程 PC 通过发出合适的指令来控制。

4.3.1　具有动态增强功能的红外测距系统㊀

本书所开发的机器人系统配置有一个具有动态测距增强功能的自主开发的基于 PIC 微控制器的红外测距系统[2]。由于红外(IR)测距传感器具有体积小、简单易用、成本低和功耗低等特点，因此广泛用于机器人的测距应用。在常规形式下，利用三角测量原理的 SHARP 红外测距广泛应用于现实生活[1]。在此，红外光从物体反射的角度取决于与物体之间的距离。在本书的机器人中，所采用的红外测距系统是基于散射线的检测开发的，在基于传统三角测量原理的情况下，可减少物体表面朝向对传感器读数的影响。通常，随着与物体之间距离的减小(即物体越近)，红外测距系统的输出电压会增大。然而该系统只能适用于死区范围之外，因为对于死区内的任何测距值，输出电压会减小而不是增大[1]。出现这种情况的原因是在死区内，传感器接收不到窄红外光束的概率增大。为提高基于红外传感器避障方法的灵敏度，在机器人系统中构建了一种自主开发的基于 PIC 微控制器的红外测距系统，而不是仅采用一个简单的红外测距传感器[2]。在此所开发的红外测距系统是采用一种可使得脉冲频率和红外能量传输持续时间逐步减小的阵列方法。目的是通过利用红外传感器系统的输出自适应地切换一个红外 LED 的开/关来减小死区范围。该系统是利用在红外传感器两侧上独立分布的两个红外源来实现测距增强的。

图 4.3 给出了采用 PIC12F683 微控制器开发的硬件系统示意图[3]。两个高

㊀ 4.3.1 节是基于 Anjan Rakshit 和 Amitava Chatterjee 于 2010 年 10 月在 IEEE 传感器期刊第 10 期上 1635 ~ 1636 页发表的论文“A microcontroller based IR range finder system with dynamic range enhanement(一种具有动态测距增强功能的基于微控制器的红外测距系统)”。

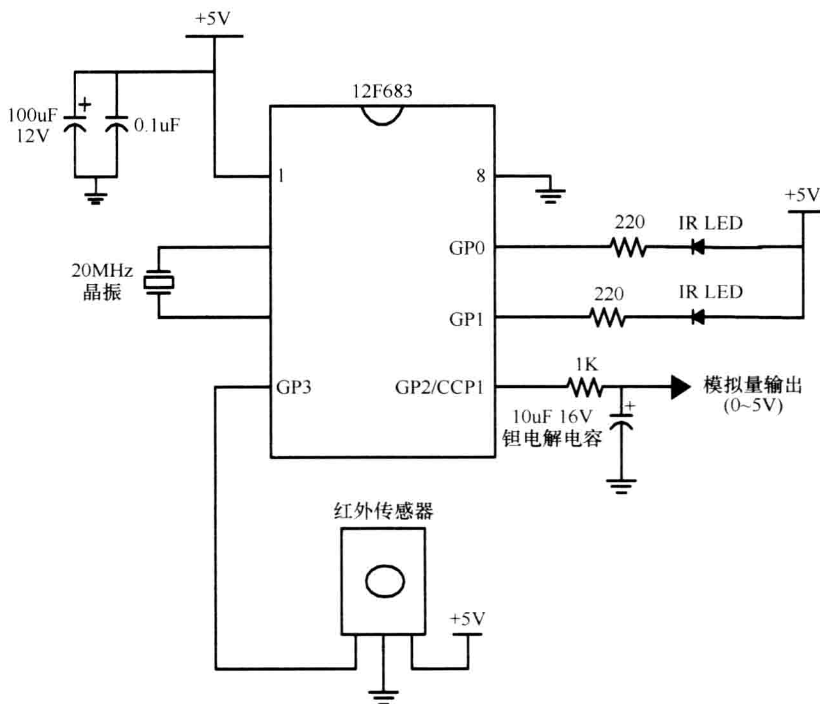


图 4.3 所开发的用于动态测距增强的基于 PIC12F683 微处理器的红外测距系统
（在得到 IEEE 的许可下，从参考文献[2]中引用）

强度的红外 LED(IR_LED1 和 IR_LED2)发出的红外能量由一个在图 4.3 中称为红外传感器(IR_Sensor)的 SHARP 红外传感器系统(IS 1U60)接收。IS 1U60 系统的内部框图[4]表明当该接收器接收输入的红外能量时，传感器的输出电压会降低，反之亦然。带通滤波器的中心频率 $f_0 = 38\text{kHz}$。出于频率调制的目的，载波频率大约为 38kHz 时的相对灵敏度最大[4]。在标称情况下，可输出持续时间为 600μs 的占空比为 50%、频率为 38kHz 的脉冲波信号[4]。

算法 4.1 给出了在 PIC 微控制器上实现的主程序。算法 4.2 给出了定时器 1 (Timer1)溢出使能时所开发的实时中断子程序，该子程序与主程序共同执行。在此引入两个数组：控制 IR_LED 载波频率或脉冲频率的 Burst_Freq_Array 和决定一次扫描中 IR_LED 传输多久的 Inegral_Cycle_Array。在传统的系统中，占空比为 50% 的脉冲频率为 38kHz，传输持续时间为 600μs，并且在范围相对较大的一个窄带宽上，传感器可得到一个灵敏结果。在此专门控制这两个变量，以使得在与距离相关的一次扫描中几个脉冲频率持续时间内，IR_SENSOR 可接收到一些从物体反射回的红外光能量。对该信息(Range_Count)进行指数平均可产生一个稳定的 PWM 信号。对于一个合理的传感器处理速度，只能建立有限长度

的阵列，由此对所开发的系统产生“距离量化”或有限解。

算法 4.1 PIC 微控制器中的主程序

开始
1. 在关闭状态下初始化 IR_LED1 和 IR_LED2。
2. 设置 Burst_Freq_Array 和 Integral_Cycle_Array。
3. 设置用于定时器 1 中断时的定时器 1 寄存器对。
4. 编制适当的 PWM 载波频率。
5. 从中断子程序接收 Range_Count 信息。
6. 对该信息扩展，以适用于 PWM 生成。
7. 利用指数平均生成 PWM 信号。
8. 返回到 5。
结束

算法 4.2　中断子程序

```
开始
1. 准备利用 Burst_Freq_Array [i]进行中断。
2. 设 Count1_max = Internal_Cycle_Array [i]。
3. IF(Count1 > Count1_max),
     切换 Burst_Duration_flag，并重置 Count1。
     IF(Burst_Duration_flag = =0),
         i 加 1;
         IF(SIGIN = =0),
             j 加 1。
         ENDIF
     ENDIF
     IF(i 达到 Burst_Freq_Array 中最后一个元素),
     Range_Count =j; 重置 i 和 j;
     ENDIF
ENDIF
4. IF(Burst_Duration_flag = =1),
     若 Burst_Freq_flag =0，则 IR_LED1 打开。
     若 Burst_Freq_flag =0 且 SIGIN =0，则 IR_LED2 打开。
         ELSE
```

IR_LED1 和 IR_LED2 全部关闭。

ENDIF

5. 切换 Burst_Freq_flag。
6. 清除 Interrupt_flag。

结束

4.3.1.1 动态测距增强算法

通过将 IR_SENSOR 的输出作为微控制器的一个反馈信号(SIGIN)来实现动态测距增强，即自适应地开/关 IR_LED2。算法 4.2 给出了由 Burst_Freq_flag 和 SIGIN 的状态来控制 IR_LED2 闪烁。一个常用的红外测距仪中，在死区范围内时，从物体反射回的大多数红外能量都无法由 IR_SENSOR 感知。在本系统中，对于距离较远的物体，大多只有 IR_LED1 闪烁。随着逐渐接近死区，IR_LED2 开始激活，这是由于 SIGIN 减小的概率较大。这种以智能方式自适应地将 IR_LED2 的激活与距离减小相对应，将有助于减小死区范围并实现所需的动态测距增强。这与常用红外测距仪的工作原理形成鲜明对比，在常用红外测距仪中，死区范围内从物体反射回来的大部分红外能量都无法由 IR_Sensor 感知。

4.3.1.2 实验结果

在本实验室进行了一个实验，即对于没有测距增强的系统，在中断服务子程序中不利用反馈信号 SIGIN 来控制 IR_LED2。图 4.4 给出了这两种情况下的输出电压随距离的变化曲线。在具有和不具有测距增强的两种情况下，在每次距离不同时，输出电压取 10 次读数的平均值。对于没有动态测距增强的系统，有效距离为 25 ~ 50cm，低于 25cm 就进入死区。由图可知，本书所开发的系统可减少死区范围，使得动态增强测距的有效距离达到 10 ~ 50cm。

4.3.2 采用开关模式同步检测技术的光学接近检测器㊀

自主研制的机器人系统还配置了基于开关模式同步检测原理在 PIC 微控制器中开发的光学接近检测器[15]。近年来，基于微控制器的系统已广泛用于开发低成本的机器人传感器系统[2]以及一些智能仪器仪表装置[5-7]。本节将介绍一种基于开关模式同步检测技术的 PIC 微控制器[3]中光学接近传感器系统的开发，并采用一种有效策略来提取受噪声严重影响的信号的基波分量。该同步检测技术已广泛应用于 AM 无线电接收器、交流偏置的应变片桥式电路、测温仪系

㊀ 4.3.2 节是基于 Anjan Rakshit 和 Amitava Chatterjee 于 2012 年 5 月在测量科学与技术期刊第 3 期上发表的论文“A microcontroller based compen sated optical proximity detector employing switching-mode synchronous detection technique(一种基于开关模式同步检测技术的微控制器补偿的光学接近检测器)”。

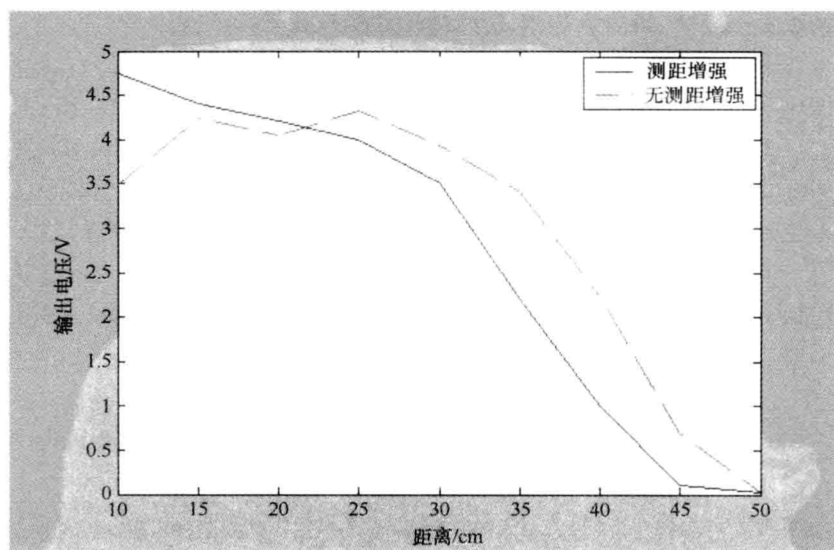


图 4.4　红外传感器系统的输出电压随距离的变化曲线

（在得到 IEEE 的许可下，从参考文献[2]中引用）

统[8]、机械振动测量仪[9]、同步相位电压转换器[10]以及光纤传感器测量仪[11]等装置中。在此，目标是开发一种成本低但功能强大的机器人传感器，并在环境照明变化范围较大的条件下，能够给出与障碍之间的准确距离。该系统采用两个发射白光的 LED 来确定与障碍之间的距离。利用一个光敏电阻(LDR)[12]和晶体管组成的电子电路来判断障碍物是否足够近。所开发的系统具有外部阈值变化的灵活性，使得在不同工作环境下，导致传感器激活的最大障碍物距离可适当变化。同时，该系统还具有动态补偿功能，使得传感器性能几乎与环境的照明条件无关。

影响接近传感器性能的因素有很多。一个物体的检测本质上取决于对从物体表面反射回的辐射检测，因此对于物体与传感器之间的距离相同，则反射回的辐射量将取决于物体的反射率。被测物体的反射率在 0 ~ 1 之间变化。高反射率的物体，其反射率接近于 1，反之则接近于 0。另一个重要的影响因素是物体的表面光滑程度，即白色 LED 光源的光能量入射物体表面的光滑度(或粗糙度)。众所周知，如果反射表面足够大，能涵盖两个 LED 发射光的整个空间分布。对于无光泽的物体，如果已知物体表面的反射率，传感器的模拟信号可用于确定与物体的距离。然而，在大多数实际情况下，机器人传感器并不清楚在其导航过程中所将要遇到的物体类型，因此物体的反射率值并不是先验已知的。考虑到这种情况，所进行的实验中设置了一组反射率变化范围较大的物体，因此所开发传感器的适用性也进行了广泛测试和验证。

4.3.2.1 基于 PIC 微控制器的光学接近检测器

图4.5给出了所开发的一种基于PIC12F675微控制器的系统。该系统具有与两个LED驱动器(LED1和LED2)相连的两路数字输出端(引脚3和引脚5)，两路模拟输入端(引脚6和引脚7)和一路控制LED(记为PXD_LED)开/关的数字输出端(引脚2)。引脚7的输入来自于PNP型晶体管的集电极，该晶体管发射极电路中包含一个阻值随光照变化的光敏电阻(LDR)[12]。在系统的输入引脚6处设计了一个用于调节阈值电压(THLD_val)的外部预设电位器，这能够有效防止PXD_LED的误激活。每个LED都由一个相同的矩形脉冲驱动，控制其在某

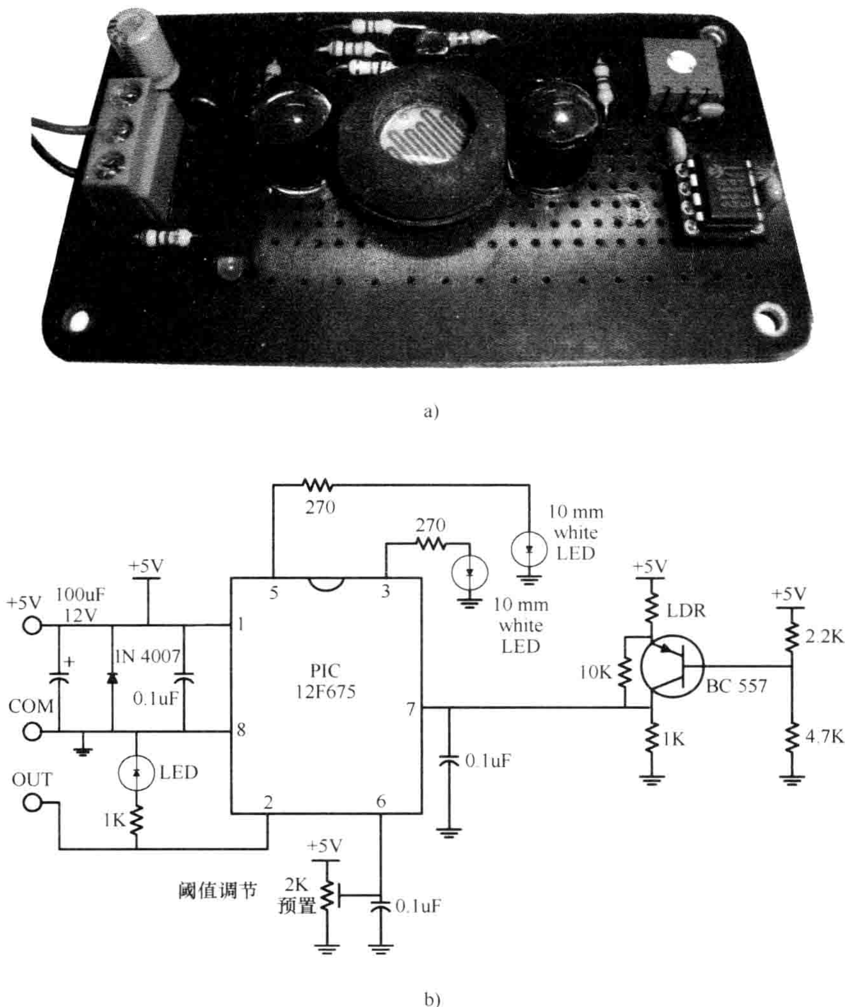


图 4.5 所开发的光学接近检测器系统

a)硬件系统 b)示意图

一给定持续时间内同时开/关(ON/OFF)。

算法4.3给出了在PIC12F675微控制器上实现的主程序。算法4.4给出了具有Timer1溢出使能且与主程序共同工作的实时中断子程序。在此，所设计的系统能在1ms的时间间隔内产生每次中断。每产生一次中断，计数器(记为Count1)值加1。根据设计思想，在驱动每个LED的矩形脉冲中，ON和OFF的持续时间均不同，在每个周期中，ON的持续时间由参数Count1_on_max控制，而在每个周期中，OFF的持续时间由参数Count1_off_max控制。在每个周期中，只要Count1的值在Count1_on_max范围内，那么LED1和LED2就都保持为ON。对于每个时间周期，当Count1的值在[Count1_on_max，Count1_off_max]范围内，则LED1和LED2均保持OFF。当这两个LED为ON时，发出白光。根据从附近物体反射回的光辐射量，可由LDR晶体管发出，在输入引脚7接收的电压信号来确定与物体之间的距离。对于每次采集到的输入信号，都要保持到一个开/关时间周期的结束，这样是为了在进行实际测量之前保持模拟信号稳定。因此，都是在(Count1 = = (Count1_on_max -2))时刻或(Count1 = = (Count1_off_max -2))时刻进行信号采集的。

对每个采集信号进行三点中值滤波来消除杂散高频分量，尤其是可能干扰原始信号的脉冲信号。对于在引脚7采集并经过中值滤波的信号，若是在LED ON时间内采集的，记为LDR_on_val。同理，同样的在引脚7采集并经过中值滤波的信号，若是在LED OFF时间内采集的，则记为LDR_off_val。不难理解，如果存在一个非常近的物体/障碍物，则LDR_on_val要比LDR_off_val大得多。因此，理论上，(LDR_on_val-LDR_off_val)的值越大意味着与物体距离越近，若(LDR_on_val-LDR_off_val)大于某一阈值，那么输出PXD_LED将变为ON，表示接近检测传感器被激活。然而根据不同的环境，也可能出现在阈值固定的某些情况下，即使物体并不非常近，PXD_LED也会变为ON。因此，为避免出现这种误激活，用户可利用外部设置一个POT来灵活调节所选阈值(采集，中值滤波处理并记为THLD_val)。由算法4.3可知，如果(LDR_on_val-LDR_off_val)大于THLD_val，则可认为接近传感器与障碍物非常接近，输出PXD_LED变为ON，否则为OFF。

该系统还采用了一种可动态应对环境光照变化的智能补偿机制。设计LDR晶体管电路使得晶体管始终保持LDR的两端电压几乎恒定，由此在不同的环境光照条件下，确保LDR具有相同的信号电平。因此，即使在环境光照最差和最佳的两种极端条件下，都能保证信号电平近似恒定。这也保证了传感器具有近似线性的灵敏度，即输出电压(即引脚7的输入电压)的增量变化与光学传感器和障碍物之间相对距离的增量变化的比值[即($\Delta V/\Delta x$)值]几乎恒定。这能直接转换为任何传感器系统设计中的一个非常重要的特性，即尽管存在其他次要因

素(本实验中是环境光照)的变化，但检测器输出电压的变化几乎是与主要被测对象(本实验中是传感器与障碍物之间的距离)的变化保持一致。

算法 4.3 PIC 微控制器中的主程序

开始

1. 设置 Timer1 为 1ms 的定时器 1 中断。

2. IF(Count1 = = (Count1_on_max-2)),

在引脚 7 接收来自 LDR 晶体管电路的输入信号。

ENDIF

3. IF(Count1 = = (Count1_on_max-1)),

在引脚 1 对 10 位 A-D 转换的模拟信号进行中值滤波，并保存为 LDR_on_val。

ENDIF

4. IF((Count1 > = Count1_on_max)&(Count1 < Count2_on_max)),

将引脚 6 接收的 THLD 作为输入信号。

对引脚 6 的 10 位 A-D 转换的模拟信号进行中值滤波，并保存为 THLD_val。

ENDIF

5. IF(Count = = (Count1_off_max-2)),

在引脚 7 接收来自 LDR 晶体管电路的输入信号。

ENDIF

6. IF(Count1 = = (Count1_off_max-1)),

在引脚 7 对 10 位 A-D 转换的模拟信号进行中值滤波，并保存为 LDR_off_val。

ENDIF

7. IF((LDR_on_val-LDR_off_val) > THLD_val),

  IF(Count2 = = Count2_max),

  将 Count2 复位为 0。

  ENDIF

  IF((Count2 > (Count2_max - 10))&(Count2 < Count2_max)),

    PXD_LED 开。

    ENDIF

  ENDIF

结束

算法 4.4 中断子程序

```
开始
1. 设置 1ms 的定时器中断。
2. Count1 加 1。
3. IF(Count2 < Count2_max),
     Count2 加 1。
     ENDIF
4. IF(Count1 > Count1_on_max),
     LED1 和 LED2 关闭。
     ENDIF
5. IF(Count1 > Count1_off_max),
     将 Count1 重置为 0。
     LED1 和 LED2 打开。
     ENDIF
结束
```

4.3.2.2　开关模式同步检测技术

同步检测技术是一种用于在强噪声中提取微弱信号基波分量的常用信号处理技术。该技术广泛应用于无线通信、工业环境(这里存在严重噪声污染或干扰信号的可能性非常大)等领域，而且该技术还需要一个频率和相位均已知的参考信号[8]。另外，同步检测技术的一个常见应用包括设计调幅信号超外差接收机。在传统的同步检测方法中，参考信号通常是一个正弦信号或谐波信号。对于这种传统技术，一种常用的形式是以方波/矩形波为参考信号的开关模式同步检测(SMSD)技术。开关模式同步检测器的核心在于采用了一个相敏检测器。在开关模式同步检测技术[13]中，一个周期性矩形脉冲序列作为参考信号 $r(t)$，用于采样噪声信号 $x(t)$，并对检测器的输出信号 $x_m(t)$ 进行低通滤波来恢复 $x(t)$ 的基波分量，即 $x_f(t)$。在本实验中，采用一种改进的开关模式同步检测技术，如图 4.6 所示。其中，矩形参考信号 $r(t)$ 用于采样噪声信号 $x(t)$，这将产生一种与传统 SMSD 方法相同的检测器输出信号 $x_m(t)$。然后，只有当输出信号 $x_m(t)$ 长时间产生一个高电平信号，才能激活 LED 的输出。这与条件采样和保持操作相类似，可等效于一个低通滤波的作用，这可避免在微控制器输入引脚 7 采集的可能由一些不必要的外部干扰所产生的高电平输入脉冲信号或短时输入信号导致激活接近探测器。如果该信号产生一个持续很长时间的高电平，则可推断这一定是由于在传感器附近存在物体造成的，而不是采集的噪声信号。图 4.7 给出了在传感器附近存在物体时的一个实际采样情况。其中，微控制器引脚 7 的采

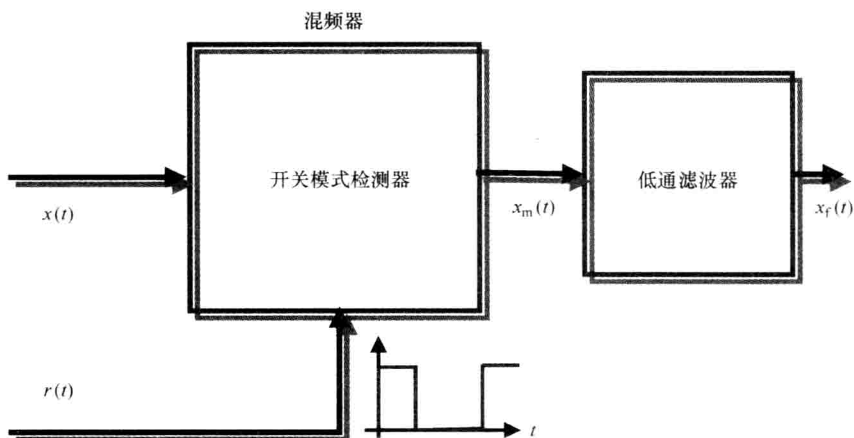


图 4.6 一种改进的开关模式同步检测器

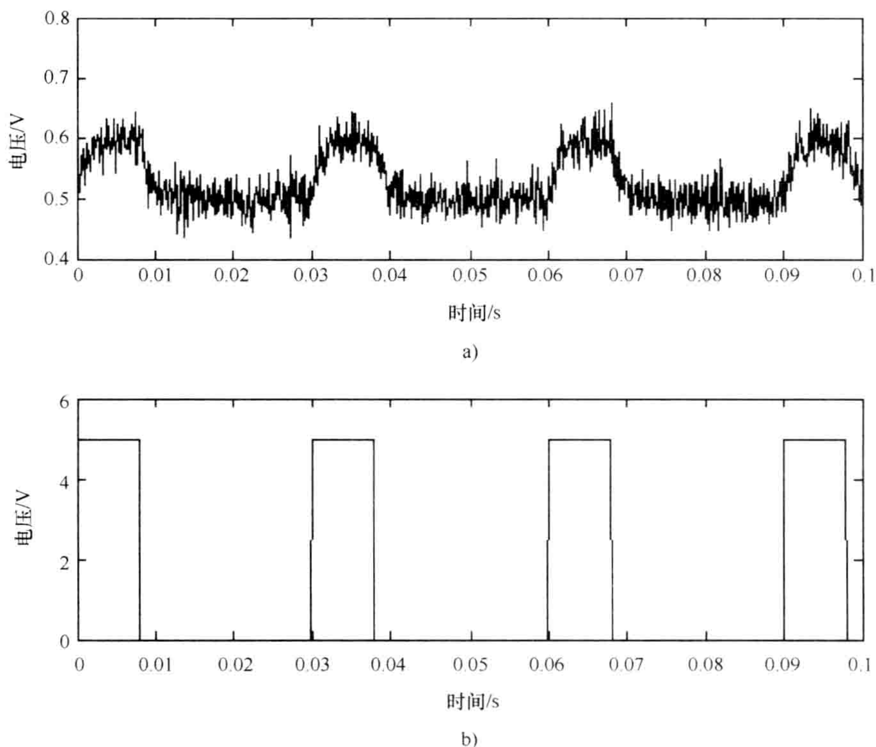


图 4.7 实际采样的输入信号 $x(t)$（见图 4.7a）和参考信号 $r(t)$（见图 4.7b）

集信号作为输入信号 $x(t)$，参考信号表示为 $r(t)$。值得注意的是，在开关模式同步检测技术中，要分析的信号与参考信号之间的相对相位差具有重要作用[8]。对于与参考信号相位不匹配的信号，若相位差为 90°，则输出信号减小，且在一

个给定频率下，对开关模式检测器输出信号没有影响。在本实验中，在引脚 7 采集的晶体管发射信号为 $x(t)$，而 LED 产生参考信号 $r(t)$。若一个物体距离足够近，则低通滤波器产生一个较高的输出信号，而对于距离较远的物体，则输出信号较低。THLD 信号作为一个布尔信号，用于判断与物体之间的距离。

4.3.2.3　实验结果

利用所设计的光学接近检测器在实际环境中的各种实验条件下来检测周围物体。每次实验，传感器系统都能通过一个布尔型输出，即 LED 的输出(在足够接近一个物体时，PXD_LED 打开，否则关闭)，来表明具有良好性能。然而如上所述，如果传感器与物体之间的相对距离不断减小，则能改变布尔型输出(即从关闭状态变为打开状态)的传感器与物体所处位置之间的最小距离取决于多种因素。图 4.8 给出了在测试中与物体之间最小距离变化的影响的实验结果，在此需要一个利用 POT 进行外部调节的阈值电压(THLD_val)的函数来激活接近检测器。正如预期的那样，随着阈值增大，一般情况下，检测器在最小接近距离更小时才会激活。若阈值设定得较高，系统会表现出一种近似饱和效应，这表明对于激活检测器的最小距离存在一个有效死区。

图 4.8 中给出了三种不同类型物体时的实验结果：①具有较高的反射率(p

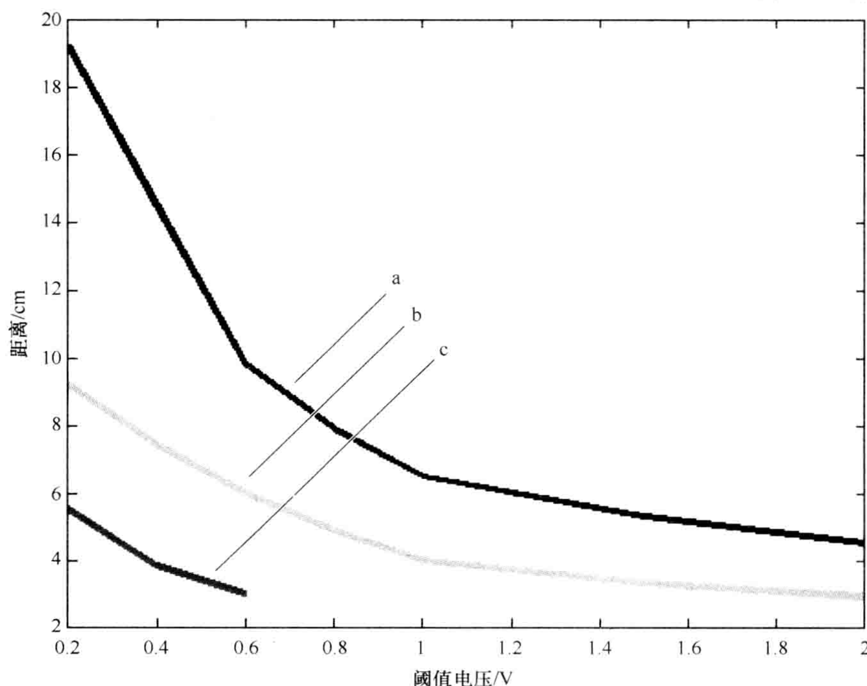
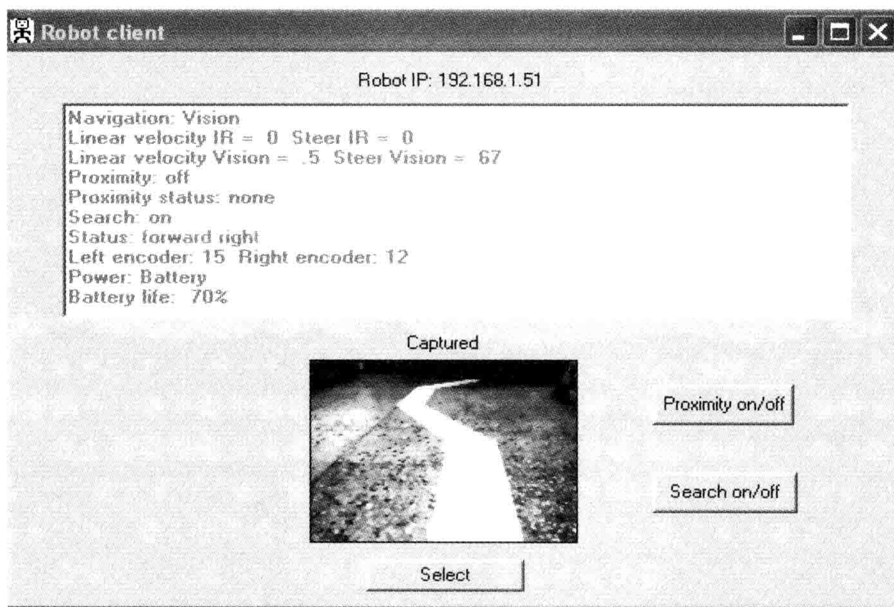


图 4.8　物体反射率分别为 $p=33\%$ (a)、$p=16\%$ (b)和 $p=7.8\%$ (c)时接近检测器的性能曲线

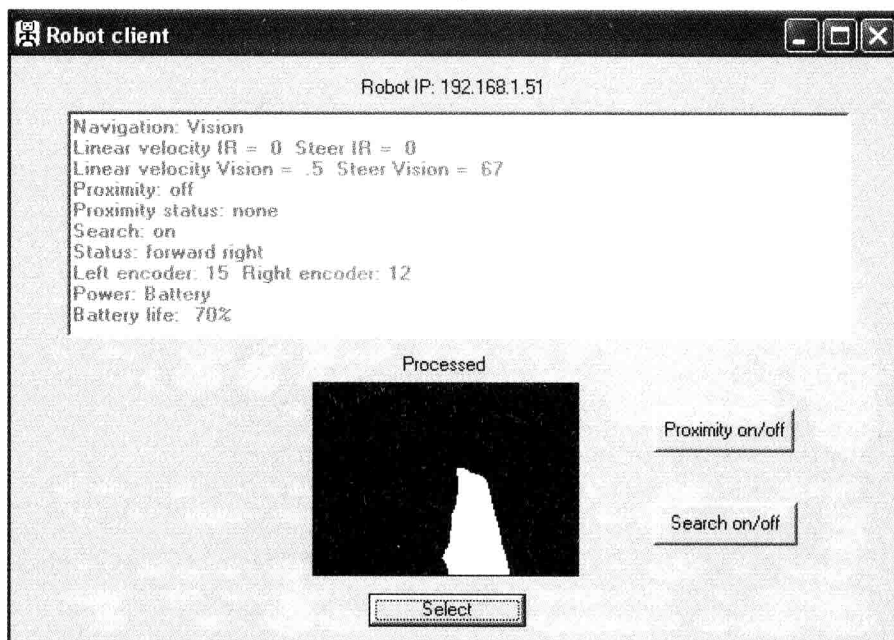
= 33%）；②具有适中的反射率（$p = 16\%$）；③具有较低的反射率（$p = 7.8\%$）。这些反射率值都是在波长为550nm处获得的。利用KYORITSU 5200型照度计，对每个物体的反射率进行实验测定。通过保持每个物体的反射表面与LED光轴正交来得到实验结果。不难看出，对于高反射率的物体，在设置某一给定阈值电压条件下，距离相对较远时即可激活接近检测器。如果换作具有低反射率的其他物体，同样的阈值电压条件下，只能在距离非常近时才能激活接近传感器，即需要能够激活接近输出LED的最短距离非常小。对于反射率较低的物体，相应的接近距离也非常小，且即使所选的阈值电压很小，传感器也很快进入死区。例如，本实验中，在阈值电压为0.6V时，可进入物体（c）的死区，若进一步增大该阈值电压，系统就不能有效地接近检测。因此，为有效利用机器人导航中的接近传感器，物体应至少具有中等或中低等的反射率，才能使得机器人在传感器激活的条件下安全避障。同时，实验也验证了该传感器系统可有效地用于检测最小为6cm×8cm以及更大的物体。

4.4 客户端—服务器工作模式下的内网连接

除了上述两种特殊类型的传感器系统外，自主研制开发的机器人还能够在机器人上安装的笔记本电脑与远端PC之间进行内网连接数据通信和控制命令交互。在这种客户端—服务器工作模式下，机器人作为服务器，而远端用户作为客户端，通过Windows环境下基于TCP / IP的Socket编程来进行通信。在客户端能够选择机器人的功能，如是否在导航过程中使用接近传感器；是否开启或关闭搜索模式。如果开启搜索模式，则在机器人偏离路径时，可利用所保存的绕行转向角信息来导引机器人返回路径。如果在客户端取消搜索模式，则机器人一旦偏离路径就会停止运动。如果需要的话，可在远程操作中增加更多控制灵活性来实现更多功能。服务器也可根据来自客户端的“数据发送请求（data send_request）”来发送文本和图像数据。通常，由于图像数据量较大，在接收到一个接收请求时，服务器首先创建一个用于传输图像矩阵的所有像素值的数组。然而，整个数据数组并不是以连续方式传输，而是通过由低端设备驱动程序管理的一系列数据包发送。同时，客户端系统也以同样的方式编程，即不断接收数据包，直到接收完一个完整的图像数据阵列，然后重建在客户端显示的图像。该系统具有互锁功能，即当客户端正在处理上一个接收数据包的请求过程中，不允许发送新的数据请求。图4.9分别给出了显示一帧采集图像和在服务器对该图像进行路径提取的客户端用户界面。根据图形用户界面的显示以及前面所述，通过在客户端点击“接近检测器 开/关”按钮来激活或停用红外接近传感器，该系统具有很大的灵活性。另外，还可通过点击“搜索模式 开启/关闭”按



a)



b)

图 4.9　机器人导航过程中在客户端显示的一个采样时刻时的 GUI 界面

钮来表明在机器人视场中路径消失时，机器人是继续以迭代方式来重新定位到路径，还是只是简单地停止进一步导航。

4.5 本章小结

本章介绍了如何在实验室中自主开发一个具有特殊功能的低成本机器人。该机器人系统包括两种专门开发的基于微控制器的传感器系统，并且通过内网灵活连接。在两种专门开发的传感器系统中，基于 PIC 微控制器的红外测距系统是利用红外传感器的输出来自适应地切换一个红外 LED 的开/关，从而实现动态测距增强。该系统采用一种基于阵列的方法来控制脉冲频率和红外能量传输的持续时间，以提高测距精度。另一种基于微控制器的传感器系统包括一个由发射白光的 LED、利用 LDR 和晶体管组成的电子电路以及一个布尔表示开/关的输出 LED 组成的光学接近传感器检测系统。该系统是采用开关模式同步检测技术来开发的，同时为了使得该电路能在不同工作条件下可靠运行，系统中还配置了一个调节外部阈值的装置，从而便于用户对系统性能进行微调。最后，该机器人在客户端—服务器工作模式下进行内网连接，其中机器人上的笔记本电脑工作在从模式，而工作在主模式下的远程 PC 可从远端给机器人发出合适的操作命令。

参考文献

[1] De Nisi, F., Gonzo, L., Gottardi, M., Stoppa, D., Simoni, A., Angelo-Beraldin, J.: A CMOS sensor optimized for laser spot-position detection. IEEE Sensors Journal 5(6), 1296–1304 (2005)

[2] Rakshit, A., Chatterjee, A.: A microcontroller based IR range finder system with dynamic range enhancement. IEEE Sensors Journal 10(10), 1635–1636 (2010)

[3] http://ww1.microchip.com/downloads/en/DeviceDoc/41211D_.pdf

[4] http://owww.phys.au.dk/elektronik/is1u60.pdf

[5] Chatterjee, A., Sarkar, G., Rakshit, A.: Neural compensation for a microcontroller based frequency synthesizer-vector voltmeter. IEEE Sensors Journal 11(6), 1427–1428 (2011)

[6] Chatterjee, A., Sarkar, G., Rakshit, A.: A reinforcement-learning-based fuzzy compensator for a microcontroller-based frequency synthesizer/vector voltmeter. IEEE Transactions on Instrumentation and Measurement 60(9), 3120–3127 (2011)

[7] Ray, S., Sarkar, G., Chatterjee, A., Rakshit, A.: Development of a microcontroller-based frequency synthesizer cum vector voltmeter. IEEE Sensors Journal 11(4), 1033–1034 (2011)

[8] Platil, A.: An introduction to synchronous detection, http://measure.feld.cvut.cz/en/system/files/files/en/education/courses/xe38ssd/SynchrDetectBW.pdf

[9] Borza, D.N.: Mechanical vibration measurement by high-resolution time-averaged digital holography. Measurement Science and Technology 16(9), 1853 (2005), doi:10.1088/0957-0233/16/9/019

[10] Ruggeri, M., Salvatori, G., Rovati, L.: Synchronous phase to voltage converter for true-phase polarimeters. Measurement Science and Technology 16(2), 569 (2005), doi:10.1088/0957-0233/16/2/033

[11] Philp, W.R., Booth, D.J., Shelamoff, A., Linthwaite, M.J.: A simple fibre optic sensor for measurement of vibrational frequencies. Measurement Science and Technology 3(6), 603 (1992), doi:10.1088/0957-0233/3/6/007

[12] PHILIPS Data handbook. Semiconductors and integrated circuits, Part 4b (December 1974)

[13] Min, M., Parve, T.: Improvement of the vector analyser based on two-phase switching mode synchronous detection. Measurement 19(2), 103–111 (1996)

[14] Nirmal Singh, N.: Vision Based Autonomous Navigation of Mobile Robots. Ph.D. Thesis, Jadavpur University, Kolkata, India (2010)

[15] Rakshit, A., Chatterjee, A.: A microcontroller based compensated optical proximity detector employing switching-mode synchronous detection technique. Measurement Science and Technology 23(3) (March 2012), http://m.iopscience.iop.org/0957-0233/23/3/035102

第5章　基于视觉的移动机器人导航算法的实现示例

摘要：本章阐述了在32位的Windows系统环境下，如何逐步详细地真正实现基于视觉的导航模块。在这些示例中，首先是一个从正在运行的视频中采集图像帧的简单示例，然后是逐步处理集成了图像处理功能（如过滤技术、对比度增强、自适应阈值等）的复杂任务。另外，这些示例还展示了如何从这些图像中提取机器人路径以及如何采用基于规则的方法来确定一个差动驱动系统的左右轮速度的设定。

5.1　简介

本章一步步地介绍了基于Visual Basic的软件编程，并给出了在PC中十个视觉导航的编程示例。在此，利用低成本的网络摄像头来采集视频流，并采用Visual Basic 6（VB6）[1,2]来进行Windows编程。

第一个示例“示例1”介绍了如何在低成本网络摄像头获取的视频流中采集图像帧并用鼠标查看像素值。另外还以全像素方式进行RGB（红—绿—蓝）到灰度的图像变换。通过“Format”菜单来选择图像帧大小为160×120。调用32位Windows API（应用程序界面）[3]来加速图像处理过程。

第二个示例“示例2”展示了如何处理从视频流中采集的图像帧。包括RGB到灰度的图像变换和低通滤波两种操作[4]。

第三个示例“示例3”给出了在光照较差的条件下基于直方图拉伸技术[4]的对比度增强方法。

第四个示例“示例4”引入了几何均值滤波器来平滑和抑制图像细节，从而简化了导航中所需白色路径的提取。

第五个示例“示例5”在光照变化条件下采用一种自适应阈值方法来提取白色路径。其中，一个可选的参考像素决定了所提取路径的中心。

第六个示例“示例6”介绍了一种清除操作来去除阈值操作中所检测到的不必要物体。

第七个示例“示例7”介绍了一种选择黑色路径或白色路径的操作。对于黑色路径，首先对灰度图像帧反相，从而使黑色物体变成白色，然后按照第六个

示例中所讨论的方法进行处理。

第八个示例“示例 8”目的是以一个固定参考像素点来确定导航所需的黑色或白色路径。

第九个示例“示例 9”介绍了一种基于规则的方法来确定一个差动驱动系统在导航过程中的左右轮速度。在此，利用一个合适的图像文件来图形化表示导航方向。

最后的第十个示例“示例 10”通过增加声音输出来在导航过程中引起注意。

适用于 Visual Basic 6 和 Visual Basic. net 2010 编译器的源代码可从 http：//extras. springer. com 下载。

若读者没有相应的编译器，本书还提供了可执行程序以用于测试程序性能。只需从微软公司免费下载所需的运行时可执行文件即可。

5.2　示例 1

目标：开发一个采集网络摄像头视频流的 VB6 程序。

以下步骤总结概括了程序的开发过程。

1）在‘Webcam1. bas’模块中声明所有必需的 API 调用。需在 VB6 程序的‘Form1’中包含这一模块。

2）AVICAP32. DLL 用于通过调用适当的 API 来采集网络摄像头的视频流。网络摄像头的视频格式应为 RGB24 或 YUY2。

3）在 FORM1 中添加两个‘Picture Box’控件，‘Picture1’用于预览 30 帧/s 的视频流，‘Picture2’用于在‘Timer1’控件下，以 10ms 的固定间隔从视频流中采集图像，并将其作为剪贴板数据。

4）在‘Form1’中添加‘Capture’和‘Close’两个命令按钮用于控制图像采集过程。这两个命令按钮分别命名为‘cmdCapture’和‘cmdClose’。

5）在‘Form1’中添加菜单项‘Format’用于设置图像大小为 160×120 像素。

6）通过鼠标查看‘Picture2’图像上的所有像素点。鼠标光标会变为十字形以便于查看像素点。

7）调用‘GetPixel’API 可获得像素点的颜色。

8）通过调用‘GetRed’、‘GetGreen’和‘GetBlue’三个函数可从‘Color’中获得红（R）、绿（G）和蓝（B）三个值。三个函数如下：GetRed = Color And 255，GetGreen = （Color And 65280）\256 和 GetBlue =（Color And 16711680）\65535。

9）添加三个命名为‘Text1’、‘Text2’和‘Text3’的文本框来显示所选像

素点的红（R）、绿（G）和蓝（B）三个8bit值。

10）两个命名为‘Text4’和‘Text5’的文本框用于监视所选像素点的‘X’、‘Y’坐标。

11）添加‘Text6’文本框来观察所选像素点由RGB通过公式gray = 0.2125 * red + 0.7154 * green + 0.0721 * blue转换为8bit灰度值。

12）添加第二个计数器‘Timer2’控件可在鼠标指针不在‘Picture2’图像框上时，10ms内清除文本框中的数据。

下面给出‘Webcam1.bas’模块的程序：

```
Global Const WS_CHILD As Long = &H40000000
Global Const WS_VISIBLE As Long = &H10000000
Global Const WM_USER = 1024
Global Const WM_CAP_ DRIVER_ CONNECT = WM_ USER + 10
Global Const WM_CAP_ SET_ PREVIEW = WM_ USER + 50
Global Const WM_CAP_ SET_ PREVIEWRATE = WM_ USER + 52
Global Const WM_CAP_ DRIVER_ DISCONNECT As Long = WM_ USER + 11
Global Const WM_CAP_ DLG_ VIDEOFORMAT As Long = WM_ USER + 41
Global Const WM_CAP_ GET_ FRAME As Long = 1084
Global Const WM_CAP_ COPY As Long = 1054
Global Const WM_CAP_SET_SCALE As Integer = WM_USER + 53
Global Const SWP_NOMOVE As Integer = 2
Global Const SWP_NOZORDER As Integer = 4
Global Const HWND_BOTTOM As Integer = 1

Declare Function SendMessage Lib " user32 " Alias " SendMessageA "_
   (ByVal hwnd As Long,ByVal wMsg As Long,ByVal wParam As_
   Long,ByVal lParam As Long) As Long
Declare Function capCreateCaptureWindow Lib " avicap32.dll " Alias_
   " capCreateCaptureWindowA "(ByVal a As String,ByVal b As Long,_
   ByVal C As Integer,B yVal d As Integer,ByVal e As Integer,_
   ByVal f As Integer,ByVal g As Long,ByVal h As Integer) As Long
Declare Function SetWindowPos Lib " user32 "(ByVal hwnd AS Long,_
   ByVal hWndInsertAfter As Long,ByVal x As Long,ByVal V AS Long,_
   ByVal cx As Long,ByVal cy As Long,ByVal wFlags As Long) As Long
Declare Function GetPixel Lib " gdi32 "(ByVal hdc As Long,_
   ByVal x As Long. ByVal y As Long) As Long
```

图 5.1 给出了'Form1'的布局。

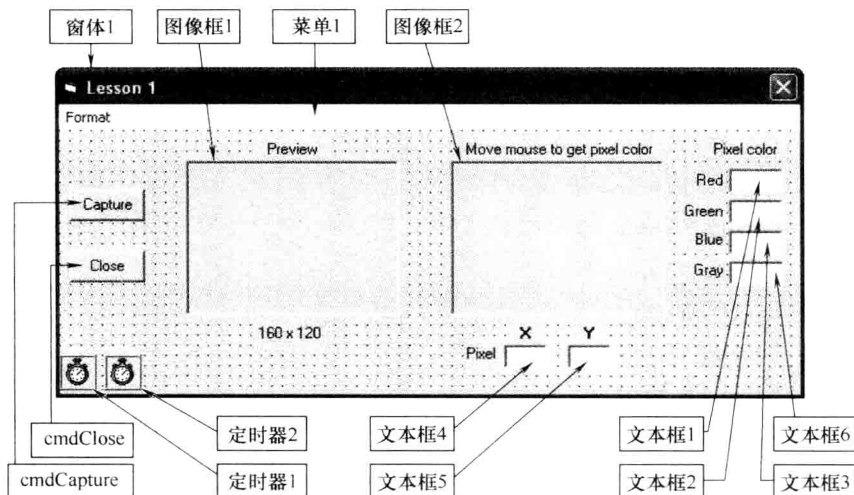


图 5.1　'Form1'的布局

以下文本显示了'Form1'的代码程序：

```
Dim hwdc As Long
Dim startcap As Boolean
Dim mflag As Boolean

Private Sub cmdCapture_Click( )
  hwdc = capCreateCaptureWindow(" Webcam Vision System ",WS_CHILD_
Or WS_VISIBLE,0,0,160,120,Picture 1. hwnd,0)
  If fhwdc < >0) Then
    Clipboard. Clear
    If SendMessage(hwdc,WM_CAP_DRIVER_CONNECT,0,0) Then
      SendMessage hwdc. WM_CAP_SET_SCALE,True,0
      SendMessage hwdc,WM_CAP_SET_PREVIEWRATE,30,0
      SendMessage hwdc,WM_CAP_SET_PREVIEW,1,0
      SetWindowPos hwdc,HWND_BOTTOM,0,0,160,120,SWP_NOMOVE_
      Or SWP_NOZORDER
      startcap = True
      cmdCapture. Enabled = Fal se
      cmdClose. Enabled = True
      Timer 1. Enabled = True
```

```
          Menu 1. Enabled = TrHe
          Picture2. Visible = True
          Label1. Visible = True
          Label2. Visible = True
          Label3. Visible = True
          Label4. Visible = True
          Label5. Visible = True
          Label6. Visible = True
          Label7. Visible = True
          Label9. Visible = True
          Label 11. Visible = True
          Text1. Visible = True
          Text2. Visible = True
          Text3. Visible = True
          Text4. Visible = True
          Text5. Visible = True
          Text6. Visible = True
      Else
          MsgBox(" No Webcam found! ")
          startcap = False
      End If
   End If
End Sub

Private Sub cmdClose Click( )
   If startcap = True Then
      SendMessage hwdc,WM_CAP_DRIVER_DISCONNECT,0. 0
      startcap = False
      cmdCapture. Enabled = True
      cmdClose. Enabled = False
      Timer1. Enabled = False
      Menu1. Enabled = False
      Picture2. Visible = False
      Label1. Visible = False
      Label2. Visible = False
```

```
        Label3. Visible = False
        Label4. Visible = False
        Label5. Visible = False
        Label6. Visible = False
        Label7. Visible = False
        Label9. Visible = False
        Label11. Visible = False
        Text1. Visible = False
        Text2. Visible = False
        Text3. Visible = False
        Text4. Visible = False
        Text5. Visible = False
        Text6. Visible = False
    End If
End Sub
Private Sub Form Load( )
    If ADp. PrevInstance = True Then End  '不允许多个实例
    cmdCapture. Enabled = True
    cmdClose. Enabled = False
    Picture1. AutoSize = True
    Picture2. AutoSize = True
    Timer1. Interval = 10
    Timer2. Interval = 10
    Menu1. Enabled = False
    mflag = False
    Picture2. Visible = False
    Picture2. MousePointer = 2              '十字光标
    Label1. Visible = False
    Label2. Visible = False
    Label3. Visible = False
    Label4. Visible = False
    Label5. Visible = False
    Label6. Visible = False
    Label7. Visible = False
    Label9. Vi sible = False
```

```
    Label11. Visible = False
    Text1. Visible = False
    Text2. Visible = False
    Text3. Visible = False
    Text4. Visible = False
    Text5. Visible = False
    Text6. Visible = False
End Sub

Private Function GetRed(ByVal Color As Long)
    GetRed = Color And 255
End Function

Private Function GetGreen(ByVal Color As Long)
    GetGreen = (Color And 65280) \256
End Function

Private Function GetBlue(ByVal Color As Long)
    GetBlue = (Color And 16711680) \65535
End Function

Private Sub Form_MouseMove(Button As Integer,Shift As Integer,_
    x As Single,y As Single)
    mflag = False                         '鼠标指针在界面中,但不在图形框中
End Sub
Private Sub Menu1_Click()
    If startcap = True Then
        SendMessage hwdc,WM_CAP_DLG_VIDEOFORMAT,0,0
    End If
End Sub

Private Sub Picture2_MouseMove(Button As Integer,Shift As Integer,_
    x As Single,y As Single)
    Dim Color As Long
    Dim red As Byte
```

```
    Dim blue As Byte
    Dim green As Byte
    Dim gray As Byte
    Dim xp As Long
    Dim yp As Long

    xp = x/Screen. TwipsPerPixelX
    yp = y/Screen. TwipsPerPixelY
    Color = GetPixel( Picture2. hdc, xp, yp)
    red = GetRed( Color)
    green = GetGreen( Color)
    blue = GetBlue( Color)
    gray =0. 2125 * red +0. 7154 * green +0. 0721 * blue
    Text1. Text = red
    Text2. Text = green
    Text3. Text = blue
    Text4. Text = xp
    Text5. Text = yp
    Text6. Text = gray
    mflag = True                          '鼠标指针在图形框中
End Sub

Private Sub Timer1_Timer( )
    SendMessage hwdc, WM_CAP_GET_FRAME,0,0
    SendMessage hwdc, WM_CAP_COPY,0,0
    Picture2. Picture = Clipboard. GetData
    SendMessage hwdc. WM_CAP_SET_PREVIEW,1,0
End Sub

Private Sub Timer2_Timer( )
    If mflag = False Then                 '鼠标指针不在图形框中
        Text1. Text = ""
        Text2. Text = ""
        Text3. Text = ""
        Text4. Text = ""
```

```
        Text5. Text = ""
        Text6. Text = ""
    End If
End Sub
```

为执行程序，需点击采集按钮。只要有网络摄像头，那么就可在图像框‘Picture1’中预览图像。如果所采集图像的大小与图像框‘Picture2’不符，则需通过激活‘Format’菜单来将图像大小转换为160×120像素。

如果没有可用的网络摄像头，则会在消息框中弹出消息“找不到任何网络摄像头！(No webcam found!)”。

5.3 示例2

目标：开发一个VB6程序来采集和处理网络摄像头的视频流，将其转换为灰度图像并进行低通图像滤波。

以下步骤总结概括了程序开发的过程：

1）在‘Webcam2.bas’模块中声明所有必需的API。另外，还需在VB6程序的‘Form1’中包含该模块。

2）AVICAP32.DLL用于通过适当的API来采集网络视频流。网络摄像头的视频格式应为RGB24或YUY2。

3）在FORM1中添加两个‘Picture Box’控件，‘Picture1’用于以10ms的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’用于在‘Timer1’控件下以同样速率对所采集图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为160×120像素。

5）调用‘GetObject’API从‘Picture1’控件获得图像像素信息。

6）在‘Timer1’控件下调用‘GetBitmapBits’API获得一个8bit数组的像素数组‘Pbytes (c, x, y)’。‘Pbytes’中每个元素均包含位于图像坐标‘x’和‘y’处的8bit RGB像素信息。‘c’代表颜色；c:2代表红色；c:1代表绿色；c:0代表蓝色。

7）根据选项‘Option1’或‘Option2’对像素数组进行处理。

8）如果选择‘Option1’，则像素数组将通过程序‘Gray’处理成灰度图像，并调用‘SetBitmapBits’API在图像框‘Picture2’中显示，如图5.2所示。

9）如果选择‘Option2’，则像素数组将首先按步骤8所示处理成灰度图像，然后调用程序‘Lowpass’进行低通滤波，最后在‘Picture2’中显示，如图5.2所示。

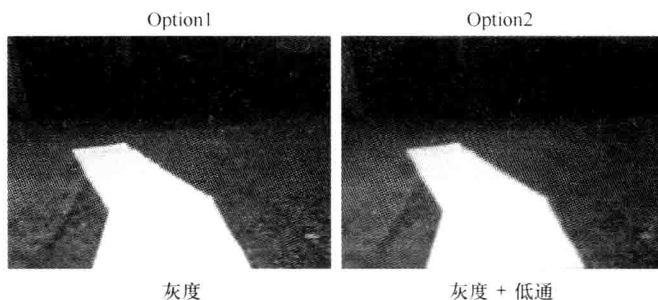


图 5.2　'Option 1' 和 'Option 2'

下面的文本给出了 'Webcam2. bas' 模块的程序：

```
Global Const WS_CHILD As Long = &H40000000
Global Const WS_VISIBLE As Long = &H10000000
Global Const WM_USER = 1024
Global Const WM_CAP_ DRIVER_ CONNECT = WM_ USER + 10
Global Const WM_CAP_ SET_ PREVIEW = WM_ USER + 50
Global Const WM_CAP_ SET_ PREVIEWRATE = WM_ USER + 52
Global Const WM_CAP_ DRIVER_ DISCONNECT As Long = WM_ USER + 11
Global Const WM_CAP_ DLG_ VIDEOFORMAT As Long = WM_ USER + 41
Global Const WM_CAP_ GET_ FRAME As Long = 1084
Global Const WM_CAP_ COPY As Long = 1054
Global Const WM_CAP_SET_SCALE As Integer = WM_USER + 53
Global Const SWP_NOMOVE As Integer = 2
Global Const SWP_NOZORDER As Integer = 4
Global Const HWND_BOTTOM As Integer = 1

Declare Function SendMessage Lib " user32 " Alias " SendMessageA "( ByVal hwnd_
  As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As_
  Long) As Long Declare Function capCreateCapturcWindow Lib_
  " avicap32. dll " Alias " capCreateCaptureWindowA "( ByVal nWindowName_
  As String, ByVal nStyle As Long, ByVal nx As Integer, ByVal ny As Integer, _
  ByVal nWidth As Integer, ByVal nHeight As Integer. ByVal nHwnd As Long, _
  ByVal nld As Integer) As Long
Declare Function SetWindowPos Lib " user32 "( ByVal hwnd As Long, _
  ByVal hWndlnsertAfter As Long, ByVal x As Long, ByVal y As Long, _
  ByVal cx As Long, ByVal cy As Long. ByVal wFlags As Long) As Long
```

```
Declare Function GetObiect Lib " gdi32 " Alias " GetObiectA "( ByVal hObject_
  As Long,ByVal nCount As Long,lpObiect As Any) As Long
Declare Function GetBitmapBits Lib " gdi32 "( ByVal hBitmap As Long,_
  ByVal dwCount As Long,lpBits As Any) As Long
Declare Function SetBitmapBits Lib " gdi32 "( ByVal hBitmap As Long,_
  ByVal dwCount As Long,lpBits As Any) As Long
```

图 5.3 给出了'Form1'的布局。

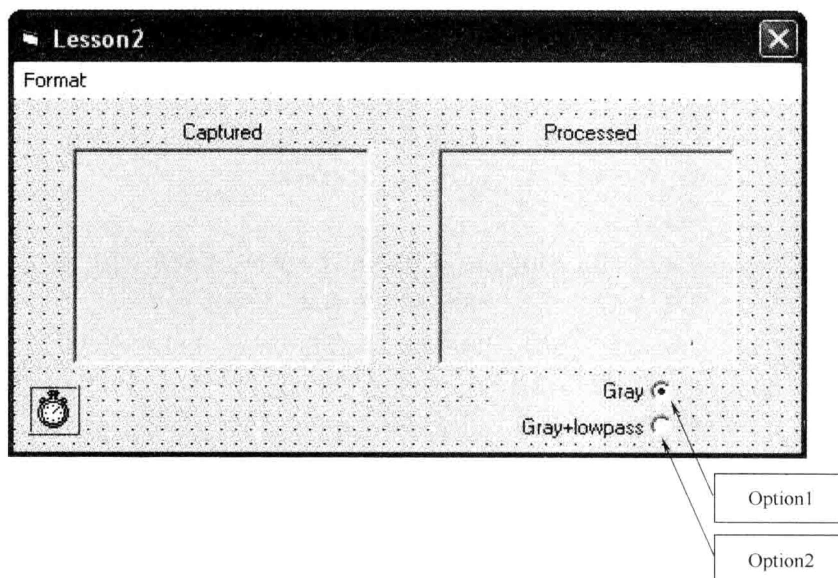


图 5.3 'Form1'的布局

以下文本给出了'Form1'的程序代码：

```
Dim hwdc As Long
Dim startcap As Boolean
Private Type Bitmap
  bmType As Long
  bmWidth As Long
  bmHeight As Long
  bmWidthBytes As Long
  bmPlanes As Integer
  bmBitsPixel As Integer
  bmBits As Long
End Type
```

```
Dim Pbytes( ) As Byte, Pinfo As Bitmap
Dim x As Long, y As Long

Private Sub Form_ Load( )
   If App. PrevInstance = True Then End
   Picture1. AutoSize = True
   Picture2. AutoSize = True
   Picture1. ScaleMode = vbPixels
   Picture2. ScaleMode = vbPixels
   Timer1. Interval = 10

hwdc = capCreateCaptureWindow( " Webcam Vision System ", WS_CHILD_
Or WS_VISIBLE, 0, 0, 160, 120, Picture 1. hwnd, 0)
If ( hwdc < > 0 ) Then
      Clipboard. Clear
      If SendMessage( hwdc, WM_CAP_DRIVER_CONNECT, 0, 0) Then
      SendMessage hwdc, WM_CAP_SET_SCALE, 1, 0
      SendMessage hwdc, WM_CAP_SET_PREVIEWRATE, 30, 0
      SendMessage hwdc, WM_CAP_SET_PREVIEW, 1, 0
      SetWindowPos hwdc, HWND_BOTTOM, 0, 0, 160, 120, _
      SWP_NOMOVE Or SWP_NOZORDER
      SendMessage hwdc, WM_CAP_GET_FRAME, 0, 0
      SendMessage hwdc, WM_CAP_COPY, 0, 0
      Picture1. Picture = Clipboard. GetData
      GetObject Picture1. Picture, Len( Pinfo ), Pinfo
      ReDim Pbytes(0 To( Pinfo. bmBitsPixel\8 ) - 1, 0 To Pinfo. bmWidth - 1, _
      0 To Pinfo. bmHeight - 1 )
      Picture2. height = Picture1. height
      Picture2. width = Picture1. width
      Timer1. Enabled = True
      startcap = True
   Else
      MsgBox " No Webcam found! ", OK. ""
      startcap = False
      Unload Me
```

```
        End If
    Else
        Unload Me
    End If
End Sub

Private Sub Gray(width As Long,height As Long)
        Dim G As Byte
        For x = 0 To width - 1
          For y = 0 To height - 1
            G = 0.2125 * CDbl(Pbytes(2,x,y)) + 0.7154 * CDbl(Pbytes(1,x,
y)) + _
              0.0721 * CDbl(Pbytes(0,x,y))
            Pbytes(2,x,y) = G            ' Red
            Pbytes(1,x,y) = G            ' Green
            Pbytes(0,x,y) = G            ' Blue
          Next y
        Next x
End Sub

Private Sub Lowpass(width As Long,height As Long)
        Dim R As Long
        Dim c,d,e,f As Long
        For x = 0 To width - 1
          For y = 0 To height - 1
          c = x - 1
          d = x + 1
          e = y - 1
          f = y + 1
          If C < 0 Then C = width - 1
          If d = width Then d = 0
          If e < 0 Then e = height - 1
          If f = height Then f = 0
          R = Pbytes(2,x,e)
          R = R + CLng(Pbytes(2,c,y))
```

```
            R = R + 2 * CLng( Pbytes(2, x, y) )
            R = R + CLng( Pbytes(2, d, y) )
            R = R + CLng( Pbytes(2, x, f) )
            R = R/6                     '3 × 3 低通
            Pbytes(2, x, y) = R
            Pbytes(1, x, y) = R
            Pbytes(0, x, y) = R
        Next y
    Next x
End Sub

Private Sub Form_Terminate( )
        If startcap = True Then
            SendMessage hwdc, WM_CAP_DRIVER_DISCONNECT, 0, 0
            startcap = False
            Timer1. Enabled = False
        End If
End Sub

Private Sub Form_Unload( Cancel As Integer)
        If startcap = True Then
            SendMessage hwdc, WM_CAP_DRIVER_DISCONNECT, 0, 0
            startcap = False
            Timer1. Enabled = False
        End If
End Sub

Private Sub Menu_Click( )
        If startcap = True Then
            SendMessage hwdc, WM_CAP_DLG_VIDEOFORMAT, 0, 0
        End If
End Sub

Private Sub Timer1_Timer( )
        Timer1. Enabled = False
```

```
        SendMessage hwdc,WM_CAP_GET_FRAME,0,0
        SendMessage hwdc,WM_CAP_COPY,0,0
        Picture 1. Picture = Clipboard. GetData
        GetBitmapBits Picture1. Picture. Pinfo. bmWidthBytes * Pinfo. bmHeight,_
          Pbytes(0,0,0)
        If Option 1. Value = True Then Gray Picture1. ScaleWidth. Picture1.  Scale-
Height
        If Option2. Value = True Then
          Gray Picture1. ScaleWidth. Picture1. ScaleHeight
          Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
        End If
        SetBitmapBits Picture2. Image,Pinfo. bmWidthBytes * Pinfo. bmHeight,_
          Pbytes(0,0,0)
        Picture2. Refresh
        Picture2. Picture = Picture2. Image
        Timer1. Enabled = True
    End Sub
```

低通滤波是通过如下所示的一个 3×3 的 2D FIR 滤波器来实现的：

$$\frac{1}{6}\begin{bmatrix}0 & 1 & 0\\ 1 & 2 & 1\\ 0 & 1 & 0\end{bmatrix}$$

对上述掩膜进行 2D 循环卷积来保存进行最小失真滤波前后的图像大小。

如果采集图像的大小与图像框不符，则需激活 Format 菜单将图像大小转换为 160×120 像素。如果没有可用的网络摄像头，则会在消息框中弹出消息“未找到任何网络摄像头！（No webcam found!）”。

5.4 示例 3

目标：开发一个 VB6 程序用于采集和处理网络视频流，并进行灰度图像转换、低通图像滤波和对比度增强处理。

以下步骤对程序开发过程进行概述：

1）与示例 2 中所述的‘Webcam2. bas’相同，所有必需的 API 调用均在‘Webcam3. bas’模块中声明。同时在 VB6 程序的‘Form1’中必须包含该模块。

2）AVICAP32. DLL 用于通过调用合适的 API 来采集网络视频流。网络摄像头视频格式应为 RGB24 或 YUY2。

3）在‘Form1’中添加两个‘Picture Box’控件，‘Picture1’用于以 10ms 的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’在‘Timer1’控件下以同样的速率对所采集的图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为 160×120 像素。

5）通过调用‘GetObject’API，从‘Picture1’中获得图像像素数据信息。

6）在‘Timer1’控件下通过调用‘GetBitmapBits’API 获得一个 8bit 的像素数组‘Pbytes（c，x，y）’。‘Pbytes’中的每个元素均包含图像坐标‘x’和‘y’处每个像素的 8bit RGB 像素信息。‘c’代表颜色；c:2 代表红色；c:1 代表绿色；c:0 代表蓝色。

7）根据选项控制‘Option1’、‘Option2’或‘Option3’来处理像素数组。

8）如果选择‘Option1’，则在程序‘Gray’下像素数组被处理成灰度图像，并调用‘SetBitmapBits’API 在图像框‘Picture2’中显示，如图 5.4 所示。



图 5.4　‘Option1’、‘Option2’和‘Option3’

9）如果选择‘Option2’，则按 8 所述，首先将像素数组处理成灰度图像，然后在程序‘Lowpass’下进行低通滤波，最后显示在‘Picture2’中，如图 5.4 所示。

10）如果选择‘Option3’，则按步骤 9 所述对数组进行低通滤波，再在程序‘Contrast’下应用直方图拉伸技术进行对比度增强，最后显示在‘Picture2’中，如图 5.4 所示。

图 5.5 给出了‘Form1’的布局。

下面的文本显示了‘Contrast’和‘Timer1’的程序代码，其余代码请参见示例 2：

```
Private Sub Contrast(width As Long,height As Long)
    Dim R As Long            '直方图拉伸
    Dim pmax,pmin As Long
    pmax = 0
    pmin = 255
```

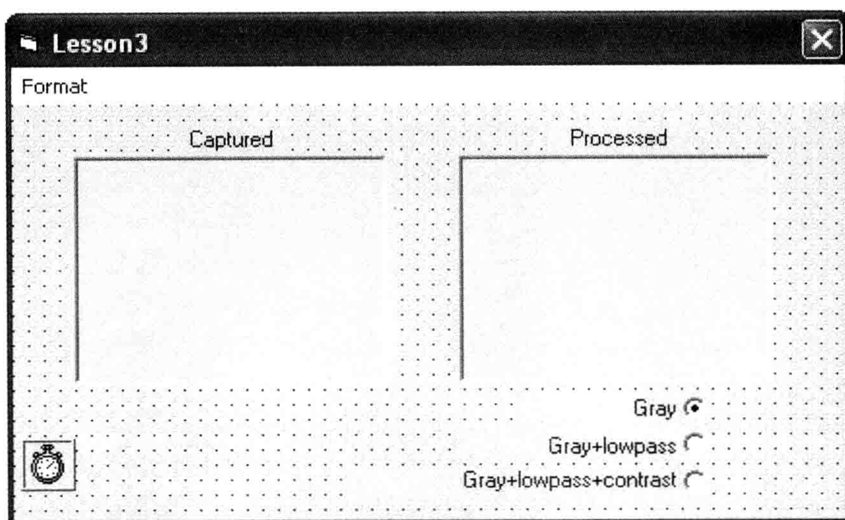


图 5.5 'Form1' 的布局

```
    For x = 0 To width - 1
      For y = 0 To height - 1
        If pmax < = CLng( Pbytes(2,x,y)) Then pmax = Pbytes(2,x,y)
        If pmin > = CLng( Pbytes(2,x,y)) Then pmin = Pbytes(2,x,y)
      Next y
    Next x
    For x = 0 To width - 1
      For y = 0 To height - 1
        R = Pbytes(2,x,y)
        If pmax > pmin Then R = (((R - pmin) * 255)/(pmax - pmin)) +
pmin/4
        If R < 0 Then R = 0
        If R > 255 Then R = 255
        Pbytes(2,x,y) = R
        Pbytes(1,x,y) = R
        Pbytes(0,x,y) = R
      Next y
    Next x
End Sub

Private Sub Timer1_Timer()
```

```
    Timer1. Enabled = False
    SendMessage hwdc,WM_CAP_GET_FRAME,0,0
    SendMessage hwdc,WM_CAP_COPY,0,0
    Picture 1. Picture = Clipboard. GetData
    GetBitmapBits Picture1. Picture,Pinfo. bmWidthBytes * Pinfo. bmHeight,_
      Pbytes(0,0,0)
    If Option1. Value = True Then Gray Picture1. ScaleWidth. Picture1. ScaleHeight
    If Option2. Value = True Then
      Gray Picture1. ScaleWidth. Picture1. ScaleHeight
      Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
    End If
    If Option3. Value = True Then
      Gray Picture1. ScaleWidth. Picturel. ScaleHeight
      Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
      Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
    End If
    SetBitmapBits Picture2. Image. Pinto. bmWidthBytes * Pinfo. bmHeight,_
      Pbytes(0,0,0)
    Picture2. Reflesh
    Picture2. Picture = Picture2. Image
    Timer1. Enabled = True
  End Sub
```

如果采集图像的大小与图像框不符，那么将激活‘Format’菜单使得图像大小转换为 160×120 像素。如果没有可用的网络摄像头，那么会在消息框中弹出一条消息“未找到任何网络摄像头！（No webcam found!）”。

5.5 示例 4

目标：开发一个 VB6 程序用于采集和处理网络视频流，进行灰度图像转换、低通图像滤波、对比度增强和几何均值滤波处理。

以下步骤对程序开发过程进行概述：

1）与示例 3 中所述的‘Webcam3. bas’相同，所有必需的 API 调用均在‘Webcam4. bas’模块中声明。同时在 VB6 程序的‘Form1’中必须包含该模块。

2）AVICAP32. DLL 用于通过调用合适的 API 来采集网络视频流。网络摄像头视频格式应为 RGB24 或 YUY2。

3）在 FORM1 中添加两个‘Picture Box’控件，‘Picture1’用于以 10ms 的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’在‘Timer1’控件下以同样的速率对所采集的图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为 160×120 像素。

5）通过调用‘GetObject’API，从‘Picture1’中获得图像像素数据信息。

6）在‘Timer1’控件下通过调用‘GetBitmapBits’API 获得一个 8bit 的像素数组‘Pbytes（c，x，y）’。‘Pbytes’中的每个元素均包含图像坐标‘x’和‘y’处每个像素的 8bit RGB 像素信息。‘c’代表颜色；c:2 代表红色；c:1 代表绿色；c:0 代表蓝色。

7）根据选项控制‘Option1’、‘Option2’、‘Option3’或“Option4”来处理像素数组。

8）如果选择‘Option1’，则在程序‘Gray’下像素数组被处理成灰度图像，并调用‘SetBitmapBits’API 在图像框‘Picture2’中显示，如图 5.6 所示。

9）如果选择‘Option2’，则按步骤 8 所述，首先将像素数组处理成灰度图像，然后在程序‘Lowpass’下进行低通滤波，最后显示在‘Picture2’中，如图 5.6 所示。

10）如果选择‘Option3’，则按步骤 9 所述对数组进行低通滤波，再在程序‘Contrast’下应用直方图拉伸技术进行对比度增强，最后显示在‘Picture2’中，如图 5.6 所示。

11）如果选择‘Option4’，则按步骤 10 所述对数组进行对比度增强，再在程序‘Geometricmean’下进行几何均值滤波，最后显示在‘Picture2’中，如图 5.6 所示。提供增加几何均值滤波器个数以及每个滤波器窗口大小的选择。

图 5.7 给出了‘Form1’的布局。

下面的文本显示了‘Geometricmean’和‘Timer1’的程序代码，其余代码请参见示例 3：

```
Private Sub Geometricmean(width As Long,height As Long,Size As Long)
    Dim R,S As Long
    Dim i,j As Long
    Dim c,d As Long
    Dim w1,h1 As Long
    If Size < 3 Then Size = 3
    If Size > 7 Then Size = 7
    If (Size And 1) = 0 Then Size = Size + 1            '奇偶转换
    S = Size * Size
```

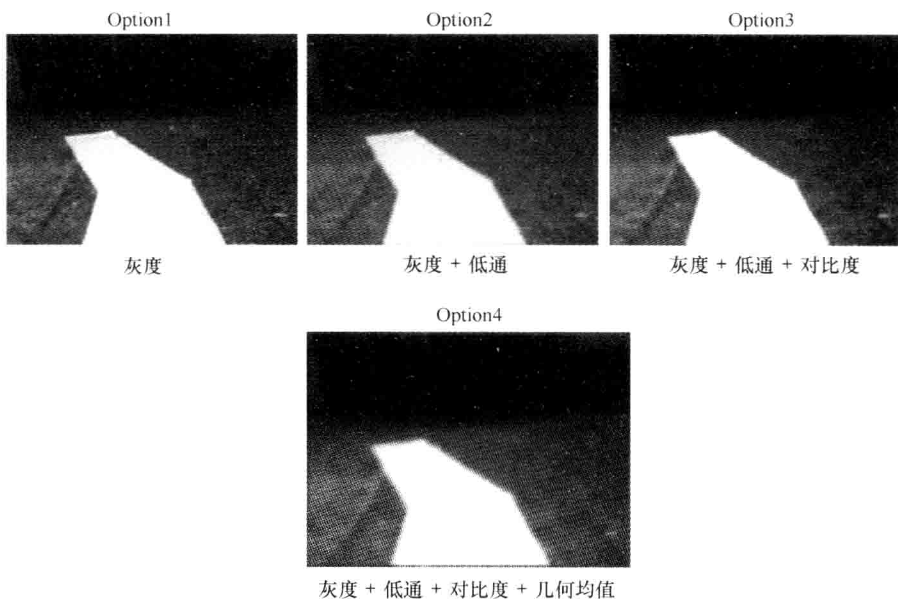


图 5.6　'Option1'、'Option2'、'Option3' 和 'Option4'

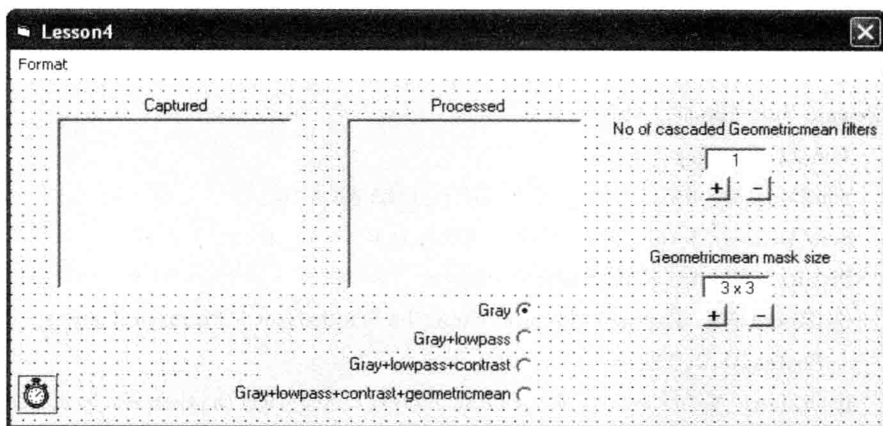


图 5.7　'Form1' 的布局

```
w1 = width - 1
h1 = height - 1
For x = 0 To w1
   For y = 0 To h1
      R = 1
      For i = 0 To Size - 1
         For j = 0 To Size - 1
```

```
                c = x + i - ((Size - 1)/2)
                If c < 0 Then c = width + C
                If c > w1 Then c = c - w1
                d = y + j - ((Size - 1)/2)
                If d < 0 Then d = height + d
                If d > h1 Then d = d - h1
                R = R * CLng(Pbytes(2,c,d))
              Next j
            Next i
            R = R^(1#/S)
            If R > 255 Then R = 255
            Pbytes(2,x,y) = R
            Pbytes(1,x,y) = R
            Pbytes(0,x,y) = R
          Next y
        Next x
End Sub

Private Sub Timer1_Timer()
    Timer1. Enabled = False
    SendMessage hwdc,WM_CAP_GET_FRAME,0,0
    SendMessage hwdc,WM_CAP_COPY.,0,0
    Picture1. Picture = Clipboard. GetData
    GetBitmapBits Picture1. Picture. Pinfo. bmWidthBytes * Pinfo. bmHeight,_
      Pbytes(0,0,0)
    If Option1. Value = True Then Gray Picture1. ScaleWidth,Picturel. ScaleHeight
    If Option2. Value = True Then
      Gray Picture1. ScaleWidth. Picture1. ScaleHeight
      Lowpass Picture1. ScaleWidth. Picturel. ScaleHeight
    End If
    If Option3. Value = True Then
      Gray Picture1. ScaleWidth. Picturel. ScaleHeight
      Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
      Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
    End If
```

```
    If Option4. Value = True Then
      Gray Picture1. ScaleWidth. Picture1. ScaleHeight
      Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
      Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
      For i = 1 To Val(Text4. Text)
        Geometricmean Picture1. ScaleWidth. Picture1. ScaleHeight. gms
      Next i
    End If
    SetBitmapBits Picture2. Image, Pinfo. bmWidthBytes * Pinfo. bmHeight, _
      Pbytes(0,0,0)
    Picture2. Refresh
    Picture2. Picture = Picture2. Image
    Timer1. Enabled = True
End Sub
```

如果采集图像的大小与图像框不符，那么将激活‘Format’菜单使得图像大小转换为160×120像素。如果没有可用的网络摄像头，那么会在消息框中弹出一条“未找到任何网络摄像头！（No webcam found!）”的消息。

5.6 示例5

目标：开发一个VB6程序用于采集和处理网络视频流，进行灰度图像转换、低通图像滤波、对比度增强、几何均值滤波和自适应阈值操作，在光照变化条件下，从采集图像中提取白色路径。

以下步骤对程序开发过程进行概述：

1）与示例4中所述的‘Webcam4. bas’相同，所有必需的API调用均在‘Webcam5. bas’模块中声明。同时在VB6程序的‘Form1’中必须包含该模块。

2）AVICAP32. DLL用于通过调用合适的API来采集网络视频流。网络摄像头视频格式应为RGB24或YUY2。

3）在FORM1中添加两个‘Picture Box’控件，‘Picture1’用于以10ms的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’在‘Timer1’控件下以同样的速率对所采集的图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为160×120像素。

5）通过调用‘GetObject’API，从‘Picture1’中获得图像像素数据信息。

6）在‘Timer1’控件下通过调用‘GetBitmapBits’API获得一个8bit的像

素数组‘Pbytes（c，x，y）’。‘Pbytes’中的每个元素均包含图像坐标‘x’和‘y’处每个像素的8bit RGB像素信息。‘c’代表颜色；c:2代表红色；c:1代表绿色；c:0代表蓝色。

7）根据选项控制‘Option1’、‘Option2’、‘Option3’、“Option4"或“Option5”来处理像素数组。

8）如果选择‘Option1’，则在程序‘Gray’下像素数组被处理成灰度图像，并调用‘SetBitmapBits’API在图像框‘Picture2’中显示。

9）如果选择‘Option2’，则按步骤8所述，首先将像素数组处理成灰度图像，然后在程序‘Lowpass’下进行低通滤波，最后显示在‘Picture2’中。

10）如果选择‘Option3’，则按步骤9所述对数组进行低通滤波，再在程序‘Contrast’下应用直方图拉伸技术进行对比度增强，最后显示在‘Picture2’中。

11）如果选择‘Option4’，则按步骤10所述对数组进行对比度增强，再在程序‘Geometricmean’下进行几何均值滤波，最后显示在‘Picture2’中。提供增加几何均值滤波器个数以及每个滤波器窗口大小的选择。

12）如果选择‘Option5’，则调用程序‘Adaptive Threshold’进行自适应阈值操作，并在‘Picture2’中显示，如图5.8所示。首先根据程序‘WhiteLineWidth’确定参考像素点［在标称位置（80，110）］附近的白色路径宽度。如果参考像素周围的左路径宽度和右路径宽度小于‘MIN_PATH_WIDTH’值，则调节参数‘delta’，通过在‘delta_max’范围内减小阈值来增大路径宽度。然后通过程序‘Threshold’来计算新图像，并重复上述操作，直到获得一个有效的白色路径。



图5.8 ‘Option5’

图5.9给出了‘Form1’的布局。

接下来的文本显示了‘AdaptiveThreshold’、“WhiteLineWidth”、“Threshold”和‘Timer1’的程序代码，其余代码请参见示例4：

```
Private Sub AdaptiveThreshol d( width As Long, xr As Long, yr As Long)
  Dim i As Integer
  WhiteLineWidth width, xr, yr
  If PixelCountLeft < MIN_PATH_WIDTH And PixelCountRight < _
    MIN_PATH_WIDTH Then
    delta = delta + 0.2
    If delta > delta_max Then
      delta = delta_max
```

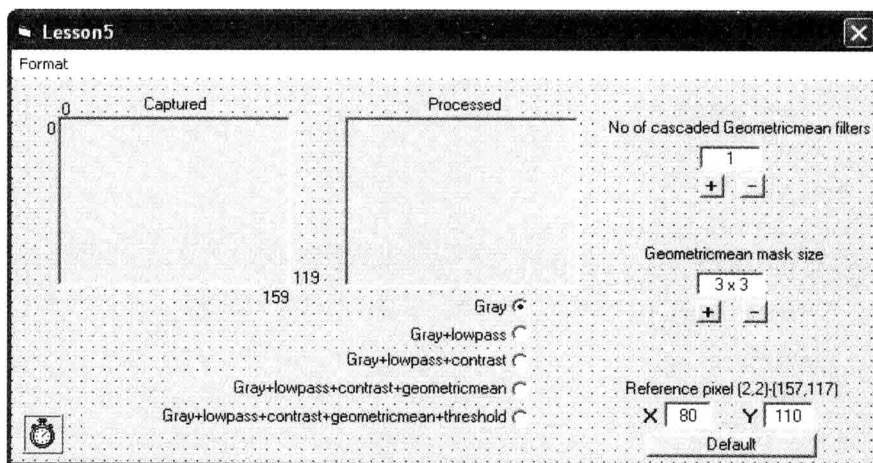


图 5.9　'Form1' 的布局

```
        Else
            GoTo atc
        End If
        If delta < 1# Then delta = 1#
    End If
        delta = delta - 0.5
atc:
    i = Pbytes(2, xr, yr)
    If i > (255 - (2 * delta)) Then
        If i > (255 - delta) Then i = (255 - delta)
        Threshold Picturel.ScaleWidth, Picture1.ScaleHeight, i - CInt(delta), _
            i + CInt(delta)
    Else
        Threshold Picture1.ScaleWidth, Picture1.ScaleHeight, 255, 255
    End If
End Sub

Private Sub WhiteLineWidth(width As Long, xr As Long, yr As Long)
    Dim pcl1, pcl2, pcl3, pcr1, pcr2, pcr3 As Integer
    PixelCountLeft = 0: PixelCountRight = 0
    y = yr
    pcl1 = 0: pcr1 = 0
```

```
    For x = xr To 0 Step - 1
      If Pbytes(2,x,y) > 250 Then
      pcl1 = pcl1 + 1
    End If
  Next x
  For x = (xr + 1) To(width - 1)
    If Pbytes(2,x,y) > 250 Then
      pcr1 = pcr1 + 1
    End If
  Next x
  y = yr - 1
  pcl2 = 0: pcr2 = 0
  For x = xr To 0 Step - 1
    If Pbytes(2,x,y) > 250 Then
        pcl2 = pcl2 + 1
      End If
    Next x
    For x = (xr + 1) To(width - 1)
      If Pbytes(2,x,y) > 250 Then
        pcr2 = pcr2 + 1
      End If
    Next x
    y = yr + 1
    pcl3 = 0: pcr3 = 0
    For x = xr To 0 Step - 1
      If Pbytes(2,x,y) > 250 Then
        pcl3 = pcl3 + 1
      End If
    Next x
    For x = (xr + 1) To(width - 1)
      If Pbytes(2,x,y) > 250 Then
        Pcr3 = pcr3 + 1
      End If
    Next x
```

```
        PixelCountLeft = ( pcl1 + pcl2 + pcl3 )/3
        PixelCountRight = ( pcr1 + pcr2 + pcr3 )/3
    End Sub
    Private Sub Threshold( width As Long,height As Long,lv As Long,hv As Long)
        Dim R As Long
        For x =0 To width - 1
          For y =0 To height - 1
            R = Pbytes(2,x,y)
            If R < 1v Then R =0
            If R > =hv Then R =255
            Pbytes(2,x,y) = R
            Pbytes(1,x,y) = R
            Pbytes(0,x,y) = R
          Next y
        Next x
    End Sub

    Private Sub Timer1_Timer( )
          Timer1. Enabled = False
          SendMessage hwdc,WM_CAP_GET_FRAME,0,0
          SendMessage hwdc,WM_CAP_COPY,0,0
          Picture1. Picture = Clipboard. GetData
          GetBitmapBits Picturel. Picture,Pinfo. bmWidthBytes * Pinfo. bmHeight,_
            Pbytes(0,0,0)
          If Optionl. Value = True Then Gray Picture1. ScaleWidth. Picture1. Scale-
Height
          If Option2. Value = True Then
            Gray Picture1. ScaleWidth. Picturel. ScaleHeight
            Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
          End If
          If Option3. Value = True Then
            Gray Picturel. ScaleWidth,Picture1. ScaleHeight
            Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
            Contrast Picture1. ScaleWidth,Picture1. ScaleHeight
          End If
```

```
        If Option4. Value = True Then
          Gray Picture1. ScaleWidth, Picture1. ScaleHeight
          Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
          Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
          For i = 1 To Val(Text4. Text)
            Geometricmean Picture1. ScaleWidth, Picture1. ScaleHeight, gms
          Next i
        End If
        If Option5. Value = True Then
          Gray Picture 1. ScaleWidth, Picture1. ScaleHeight
          Lowpass Picture1. ScaleWidth, Picture1. ScaleHeight
          Contrast Picture1. ScaleWidth, Picture1. ScaleHeight
          For i = 1 To Val(Text4. Text)
            Geometricmean Picture1. ScaleWidth, Picture1. ScaleHeight, gms
          Next i
            AdaptiveThreshold  Picture1. ScaleWidth,  Val  ( Text2. Text ),  Val
(Text3. Text)
      End If
      SetBitmapBits Picture2. Image. Pintb. bmWidthBytes * Pinfo. bmHeight,_
        Pbytes(0,0,0)
      Picture2. Refresh
      Picture2. Picture = Picture2. Image
       Picture2. Line ( Val ( Text2. Text )  - 2, Val ( Text3. Text )  - 2 )  - ( Val
(Text2. Text) +2,_
        Val(Text3. Text) +2),RGB(255,0,0),B
      Timer1. Enabled = True
    End Sub
```

如果采集图像的大小与图像框不符，那么将激活‘Format’菜单使得图像大小转换为 160 × 120 像素。如果没有可用的网络摄像头，那么会在消息框中弹出一条“未找到任何网络摄像头！（No webcam found!）”的消息。

5.7 示例 6

目标：开发一个 VB6 程序用于采集和处理网络视频流，进行灰度图像转换、低通图像滤波、对比度增强、几何均值滤波、自适应阈值以及清除操作，在光

照变化条件下，从采集图像中提取白色路径并清除不必要物体。

以下步骤对程序开发过程进行概述：

1）与示例 5 中所述的‘Webcam5. bas’相同，所有必需的 API 调用均在‘Webcam6. bas’模块中声明。同时在 VB6 程序的‘Form1’中必须包含该模块。

2）AVICAP32. DLL 用于通过调用合适的 API 来采集网络视频流。网络摄像头视频格式应为 RGB24 或 YUY2。

3）在 FORM1 中添加两个‘Picture Box’控件，‘Picture1’用于以 10ms 的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’在‘Timer1’控件下以同样的速率对所采集的图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为 160 × 120 像素。

5）通过调用‘GetObject’API，从‘Picture1’中获得图像像素数据信息。

6）在‘Timer1’控件下通过调用‘GetBitmapBits’API 获得一个 8bit 的像素数组‘Pbytes（c，x，y）’。‘Pbytes’中的每个元素均包含图像坐标‘x’和‘y’处每个像素的 8bit RGB 像素信息。‘c’代表颜色；c:2 代表红色；c:1 代表绿色；c:0 代表蓝色。

7）根据选项控制‘Option1’、‘Option2’、‘Option3’、“Option4”、“Option5”或“Option6”来处理像素数组。

8）如果选择‘Option1’，则在程序‘Gray’下像素数组被处理成灰度图像，并调用SetBitmapBits’API 在图像框‘Picture2’中显示。

9）如果选择‘Option2’，则按步骤 8 所述，首先将像素数组处理成灰度图像，然后在程序‘Lowpass’下进行低通滤波，最后显示在‘Picture2’中。

10）如果选择‘Option3’，则按步骤 9 所述对数组进行低通滤波，再在程序‘Contrast’下应用直方图拉伸技术进行对比度增强，最后显示在‘Picture2’中。

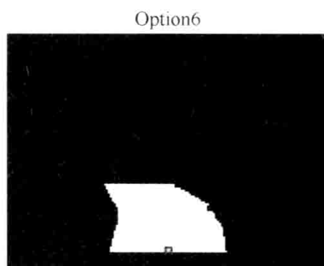
11）如果选择‘Option4’，则按步骤 10 所述对数组进行对比度增强，再在程序‘Geometricmean’下进行几何均值滤波，最后显示在‘Picture2’中。提供增加几何均值滤波器个数以及每个滤波器窗口大小的选择。

12）如果选择‘Option5’，则调用程序‘Adaptive Threshold’进行自适应阈值操作，并在‘Picture2’中显示。首先根据程序‘WhiteLineWidth’确定参考像素点［在标称位置（80，110）］附近的白色路径宽度。如果参考像素周围的左路径宽度和右路径宽度小于‘MIN_PATH_WIDTH’值，则调节参数‘delta’，通过在‘delta_max’范围内减小阈值来增大路径宽度。然后通过程序‘Threshold’来计算新图像，并重复上述操作，直到获得一个有效的白色路径。

13）如果选择‘Option6’，则调用程序‘Clean’，执行一个附加的清除操作来去除不必要物体，并在‘Picture2’中显示，如图 5. 10 所示。

图 5.11 给出了'Form1'的布局。

接下来的文本显示了'Clean'和'Timer1'的程序代码，其余代码请参见示例 5：



灰度 + 低通 + 对比度 + 几何均值 + 阈值 + 清除

图 5.10 'Option6'

```
Private Sub Clean(width As Long, height As Long,yr As Long)
        Dim R. xr,xref,xwidth As Long
        Dim PB As Long
        Dim bl_flag As Boolean
```

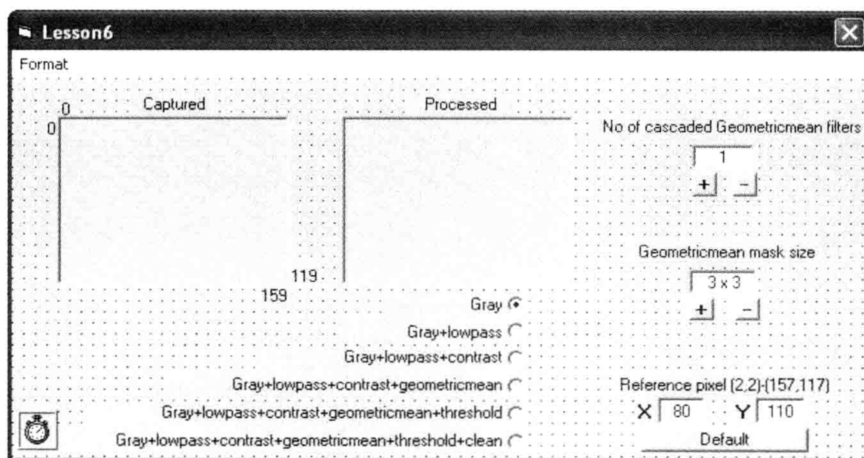


图 5.11 'Form1'的布局

```
        bl_flag = False
        xref = 0
        xwidth = 0
        If PixelCountLeft > = MIN_PATH_WIDTH Or PixelCountRight > = _
        MIN_PATH_WIDTH Then
        For x = 0 To width - 1
          R = Pbytes(2,x,yr)
          If R > 240 Then
            If xref = 0 Then xref = x
          End If
          If R > 240 And xref > 0 Then xwidth = xwidth + 1
        Next x
        xr = xref + (xwidth/2)
```

```
    For y = height - 1 To(yr + 1)Step - 1
      For x = 0 To width - 1
        Pbytes(2,x,y) = 0
        Pbytes(1,x,y) = 0
        Pbytes(0,x,y) = 0
      Next x
    Next y
    For y = yr To 0 Step - 1
        For x = xr To 0 Step - 1
          R = Pbytes(2,x,y)
          If bl_flag = True Then GoTo m1
          If R < 240 Then
            PB = x
            If PB = xr Then bl_flag = True
            GoTo m1
          End If
        Next x
m1:
        For x = PB To 0 Step - 1
          Pbytes(2,x,y) = 0
          Pbytes(1,x,y) = 0
          Pbytes(0,x,y) = 0
        Next x

        For x = (xr + 1)To width - 1
          R = Pbytes(2,x,y)
          If bl_flag = True Then GoTo m2
          If R < 240 Then
            PB = x
            If PB = (xr + 1) Then bl_flag = True
            GoTo m2
          End If
        Next x
m2:
        For x = PB To width - 1
```

```
            Pbytes(2,x,y) =0
            Pbytes(1,x,y) =0
            Pbytes(0,x,y) =0
        Next x

        xref = 0
        xwidth = 0
        For x = 0 To width - 1
            R = Pbytes(2,x,y)
            If R > 240 Then
                If xref = 0 Then xref = x
            End If
            If R > 240 And xref > 0 Then xwidth = xwidth + 1
        Next x
        If xwidth = 0 Then bl_flag = True
        For x = 0 To width - 1
            If bl_flag = True Then
                Pbytes(2,x,y) =0
                Pbytes(1,x,y) =0
                Pbytes(0,x,y) =0
            End If
        Next x
      Next y
    End If
End Sub

Private Sub Timer1_Timer()
    Timer1.Enabled = False
    SendMessage hwdc,WM_CAP_GET_FRAME,0,0
    SendMessage hwdc,WM_CAP_COPY,0,0
    Picture1.Picture = Clipboard.GetData
    GetBitmapBits Picture1.Picture.Pinfo.bmWidthBytes * Pinfo.bmHeight,_
      Pbytes(0,0,0)
    If Option1.Value = True Then Gray Picture1.ScaleWidth, Picture1. Scale-
Height
```

```
If Option2. Value = True Then
   Gray Picture1. ScaleWidth. Picture1. ScaleHeight
   Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
End If
If Option3. Value = True Then
   Gray Picture1. ScaleWidth. Picture1. ScaleHeight
   Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
   Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
End If
If Option4. Value = True Then
   Gray Picture1. ScaleWidth. Picture1. ScaleHeight
   Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
   Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
   For i = 1 To Val( Text4. Text)
      Geometricmean Picture1. ScaleWidth. Picture1. ScaleHeight,gms
   Next i
End If
If Option5. Value = True Then
   Gray Picture1. ScaleWidth,Picture1. ScaleHeight
   Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
   Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
   For i = 1 To Val( Text4. Text)
      Geometricmean Picture1. SealeWidth,Picture1. ScaleHeight,gms
   Next i
   AdaptiveThreshold Picture1. ScaleWidth, Val ( Text2. Text ), Val ( Text3.
Text)
End If
If Option6. Value = True Then
   Gray Picture1. ScaleWidth. Picture1. ScaleHeight
   Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
   Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
   For i = 1 To Val( Text4. Text)
   Geometricmean Picture1. ScaleWidth,Picture1. ScaleHeight,gms
Next i
AdaptiveThreshold Picture1. ScaleWidth,Val( Text2. Text),
```

```
        Val(Text3. Text)
      Clean Picture1. ScaleWidth,Picture1. ScaleHeight,Val(Text3. Text)
    End If
    SetBitmapBits Picture2. Image,Pinto. bmWidthBytes * Pinfo. bmHeight,_
      Pbytes(0,0,0)
    Picture2. Refresh
    Picture2. Picture = Picture2. Image
    Picture2. Line ( Val ( Text2. Text ) - 2, Val ( Text3. Text ) - 2 ) - ( Val
(Text2. Text)_
      +2,Val(Text3. Text) +2),RGB(255,0,0),B
    Timer1. Enabled = True
  End Sub
```

如果采集图像的大小与图像框不符，那么将激活‘Format’菜单使得图像大小转换为 160×120 像素。如果没有可用的网络摄像头，那么会在消息框中弹出一条“未找到任何网络摄像头！(No webcam found!)”的消息。

5.8 示例 7

目标：开发一个 VB6 程序用于采集和处理网络视频流，进行灰度图像转换、低通图像滤波、对比度增强、几何均值滤波、自适应阈值、清除操作，并在基于视觉的导航中选择黑色/白色路径。

以下步骤对程序开发过程进行概述：

1）与示例 6 中所述的‘Webcam6. bas’相同，所有必需的 API 调用均在‘Webcam7. bas’模块中声明。同时在 VB6 程序的‘Form1’中必须包含该模块。

2）AVICAP32. DLL 用于通过调用合适的 API 来采集网络视频流。网络摄像头视频格式应为 RGB24 或 YUY2。

3）在 FORM1 中添加两个‘Picture Box’控件，‘Picture1’用于以 10ms 的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’在‘Timer1’控件下以同样的速率对所采集的图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为 160×120 像素。

5）通过调用‘GetObject’API，从‘Picture1’中获得图像像素数据信息。

6）在‘Timer1’控件下通过调用‘GetBitmapBits’API 获得一个 8bit 的像素数组‘Pbytes（c，x，y）’。‘Pbytes’中的每个元素均包含图像坐标‘x’和‘y’处每个像素的 8bit RGB 像素信息。‘c’代表颜色；c:2 代表红色；c:1 代

表绿色；c:0 代表蓝色。

7）点击‘cmdWhiteBlack’按钮，在‘Shape1’中显示路径颜色（白色或黑色）。

8）如果根据步骤 7 选择黑色路径，则调用程序‘Negative’，对采集图像进行反相操作。然后根据如示例 6 所述的操作选择（“Option1”～“Option6”），对图像进行处理。

图 5.12 给出了‘Form1’的布局。

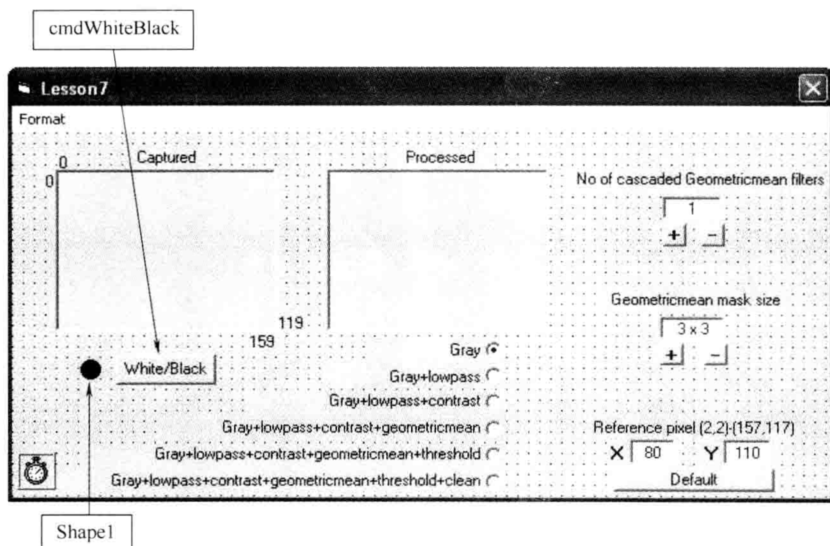


图 5.12　‘Form1’的布局

接下来的文本显示了‘cmdWhiteBlack’、‘Negative’和‘Timer1’的程序代码，其余代码请参见示例 6：

```
Private Sub cmdWhiteBlack_Click( )
    If sflag = False Then
      sflag = True
    Else
      sflag = False
    End If
    If sflag = False Then Shape1. FillColor = vbWhite
    If sflag = True Then Shape1. FillColor = vbBlack
End Sub


Private Sub Negative( width As Long, height As Long)
```

```
    Dim R As Long
    For x = 0 To width - 1
      For y = 0 To height - 1
        R = 255 - Pbytes(2,x,y) ' Invert
        Pbytes(2,x,y) = R
        Pbytes(1,x,y) = R
        Pbytes(0,x,y) = R
      Next y
    Next x
  End Sub

  Private Sub Timer1_Timer()
    Timer1. Enabled = False
    SendMessage hwdc,WM_CAP_GET_FRAME,0,0
    SendMessage hwdc,WM_CAP_COPY,0,0
    Picture1. Picture = Clipboard. GetData
    GetBitmapBits Picture1. Picture,Pinfo. bmWidthBytes * Pinfo. bmHeight,_
      Pbytes(0,0,0)
    If Option1. Value = True Then
      Gray Picture1. ScaleWidth. Picture1. ScaleHeight
      If sflag = True Then Negative Picture1. ScaleWidth. Picture1. ScaleHeight
    End If
    If Option2. Value = True Then
      Gray Picture1. ScaleWidth. Picture1. ScaleHeight
      If sflag = True Then Negative Picture1. ScaleWidth,Picture1. ScaleHeight
      Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
    End If
    If Option3. Value = True Then
      Gray Picture1. ScaleWidth,Picture1. ScaleHeight
      If sflag = True Then Negative Picture1. ScaleWidth. Picture1. ScaleHeight
      Lowpass Picture1. ScaleWidth,Picture1. ScaleHeight
      Contrast Picture1. ScaleWidth,Picture1. ScaleHeight
    End If
    If Option4. Value = True Then
      Gray Picture1. ScaleWidth,Picture1. ScaleHeight
```

```
    If sflag = True Then Negative Picture1. ScaleWidth. Picture1. ScaleHeight
    Lowpass Picture1. ScaleWidth, Picture1. ScaleHeight
    Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
    For i = 1 To Val(Text4. Text)
      Geometricmean Picture1. ScaleWidth, Picture1. ScaleHeight, gms
    Next i
End If
If Option5. Value = True Then
    Gray Picture1. ScaleWidth, Picturel. ScaleHeight
    If sflag = True Then Negative Picture1. ScaleWidth. Picture1. ScaleHeight
    Lowpass Picture1. ScaleWidth, Picture1. ScaleHeight
    Contrast Picture1. ScaleWidth, Picture1. ScaleHeight
    For i = 1 To Val(Text4. Text)
      Geometricmean Picture1. ScaleWidth, Picture1. ScaleHeight, gms
    Next i
    AdaptiveThreshold Picture1. ScaleWidth, Val(Text2. Text), Val(Text3. Text)
End If
If Option6. Value = True Then
    Gray Picture1. ScaleWidth. Picture1. ScaleHeight
    If sflag = True Then Negative Picture1. ScaleWidth. Picture1. ScaleHeight
    Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
    Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
    For i = 1 To Val(Text4. Text)
      Geometricmean Picture1. ScaleWidth, Picture1. ScaleHeight, gms
    Next i
    AdaptiveThreshold Picture1. ScaleWidth, Val(Text2. Text), _
      Val(Text3. Text)
    Clean Picture1. ScaleWidth, Picturel. ScaleHeight, Val(Text3. Text)
End If
SetBitmapBits Picture2. Image, Pinfo. bmWidthBytes * Pinfo. bmHeight, _
    Pbytes(0,0,0)
Picture2. Refresh
Picture2. Picture = Picture2. Image
Picture2. Line(Val(Text2. Text) - 2, Val(Text3. Text) - 2) _
    - (Val(Text2. Text) + 2, Val(Text3. Text) + 2), RGB(255,0,0), B
```

```
    Timer1. Enabled = True
End Sub
```

如果采集图像的大小与图像框不符，那么将激活‘Format’菜单使得图像大小转换为 160×120 像素。如果没有可用的网络摄像头，那么会在消息框中弹出一条“未找到任何网络摄像头！(No webcam found!)”的消息。

5.9 示例 8

目标：开发一个 VB6 程序用于采集和处理网络视频流，在基于视觉的导航中选择黑色/白色路径。根据路径是否存在来进行推理。

以下步骤对程序开发过程进行概述：

1）与示例 7 中所述的‘Webcam7. bas’相同，所有必需的 API 调用均在‘Webcam8. bas’模块中声明。同时在 VB6 程序的‘Form1’中必须包含该模块。

2）AVICAP32. DLL 用于通过调用合适的 API 来采集网络视频流。网络摄像头视频格式应为 RGB24 或 YUY2。

3）在 FORM1 中添加两个‘Picture Box’控件，‘Picture1’用于以 10ms 的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’在‘Timer1’控件下以同样的速率对所采集的图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为 160×120 像素。

5）通过调用‘GetObject’API，从‘Picture1’中获得图像像素数据信息。

6）在‘Timer1’控件下通过调用‘GetBitmapBits’API 获得一个 8bit 的像素数组‘Pbytes (c, x, y)’。‘Pbytes’中的每个元素均包含图像坐标‘x’和‘y’处每个像素的 8bit RGB 像素信息。‘c’代表颜色；c:2 代表红色；c:1 代表绿色；c:0 代表蓝色。

7）点击‘cmdWhiteBlack’按钮，在‘Shape1’中显示路径颜色（白色或黑色）。

8）如果根据步骤 7 选择黑色路径，则调用程序‘Negative’，对采集图像进行反相操作。然后，根据如示例 7 所述的‘Option6’，对图像进行处理。

9）根据程序‘WhiteLineWidth’来确定固定参考像素点［在（80，110）位置处］周围的白色路径宽度。如果参考像素点周围的左路径宽度和右路径宽度小于‘MIN_PATH_WIDTH’值，则推断出‘No Path’，否则得出‘Path found’的结论，并在文本框中显示。

图 5. 13 给出了‘Form1’的布局。

接下来的文本显示了‘Timer1’的程序代码，其余代码请参见示例 7：

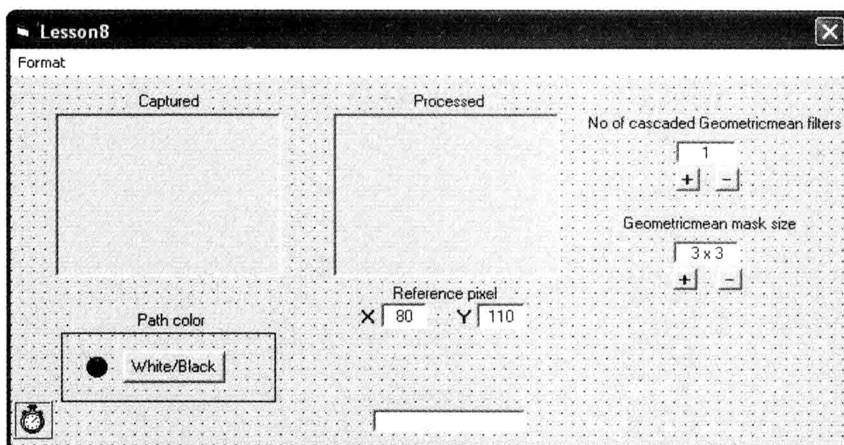


图 5.13 'Form1' 的布局

```
Private Sub Timer1_ Timer( )
    Timer1. Enabled = False
    SendMessage hwdc,WM_CAP_GET_FRAME,0,0
    SendMessage hwdc,WM_CAP_COPY,0,0
    Picture1. Picture = Clipboard. GetData
    GetBitmapBits Picture1. Picture,Pinfo. bmWidthBytes * Pinfo. bmHeight,_
    Pbytes(0,0,0)
Gray Picture1. ScaleWidth. Picture1. ScaleHeight
If sflag = True Then Negative Picture1. ScaleWidth. Picture1. ScaleHeight
Lowpass Picture1. ScaleWidth. Picture1. ScaleHeight
Contrast Picture1. ScaleWidth. Picture1. ScaleHeight
For i = 1 To Val(Text4. Text)
    Geometricmean Picture1. ScaleWidth,Picture1. ScaleHeight,gms
Next i
AdaptiveThreshold Picture1. ScaleWidth,Val(Text2. Text),Val(Text3. Text)
Clean Picture1. ScaleWidth. Picture1. ScaleHeight. Val(Text3. Text)
WhiteLineWidth Picture1. ScaleWidth,Val(Text2. Text),Val(Text3. Text)
SetBitmapBits Picture2. Image,Pinfo. bmWidthBytes * Pinfo. bmHeight,_
    Pbytes(0,0,0)
Picture2. Refresh
Picture2. Picture = Picture2. Image
Picture2. Line ( Val ( Text2. Text ) - 2, Val ( Text3. Text ) - 2 ) - ( Val
(Text2. Text)_ +2,Val(Text3. Text) +2),RGB(255,0,0),B
```

```
    If PixelCountLeft < MIN_PATH_WIDTH And PixelCountRight < _
      MIN_PATH_WIDTH Then
      Text5. Text = " No path "
    Else
      Text5. Text = " Path found "
    End If
    Timer1. Enabled = True
End Sub
```

如果采集图像的大小与图像框不符，那么将激活‘Format’菜单使得图像大小转换为 160×120 像素。如果没有可用的网络摄像头，那么会在消息框中弹出一条“未找到任何网络摄像头！（No webcam found!）”的消息。

5.10 示例 9

目标：开发一个 VB6 程序用于采集和处理网络视频流，在基于视觉的导航中选择黑色/白色路径。根据路径是否存在来进行推理。应用合适的规则来确定不同的导航方向以及差动驱动的速度参数。

以下步骤对程序开发过程进行概述：

1）与示例 8 中所述的‘Webcam8. bas’相同，所有必需的 API 调用均在‘Webcam9. bas’模块中声明。同时在 VB6 程序的‘Form1’中必须包含该模块。

2）AVICAP32. DLL 用于通过调用合适的 API 来采集网络视频流。网络摄像头视频格式应为 RGB24 或 YUY2。

3）在 FORM1 中添加两个‘Picture Box’控件，‘Picture1’用于以 10ms 的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’在‘Timer1’控件下以同样的速率对所采集的图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为 160×120 像素。

5）通过调用‘GetObject’API，从‘Picture1’中获得图像像素数据信息。

6）在‘Timer1’控件下通过调用‘GetBitmapBits’API 获得一个 8bit 的像素数组‘Pbytes（c，x，y）’。‘Pbytes’中的每个元素均包含图像坐标‘x’和‘y’处每个像素的 8bit RGB 像素信息。‘c’代表颜色；c:2 代表红色；c:1 代表绿色；c:0 代表蓝色。

7）点击‘cmdWhiteBlack’按钮，在‘Shape1’中显示路径颜色（白色或黑色）。

8）根据示例 8 处理采集图像。如果找到路径，则根据三条规则确定合适的

导航方向（‘forward’或‘turn-left’或‘turn-right’）和相应的差动驱动速度参数。图像框显示导航方向。

图 5.14 给出了‘Form1’的布局。

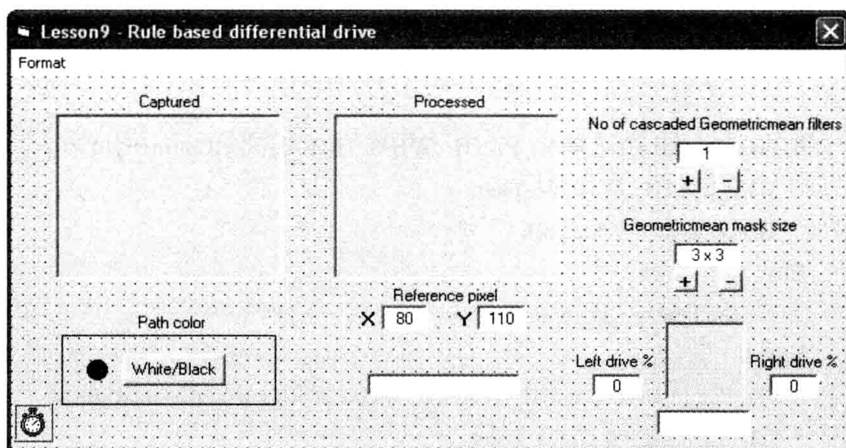


图 5.14　‘Form1’的布局

接下来的文本显示了‘Timer1’的程序代码，其余代码请参见示例 8：

```
Private Sub Timer1_Timer( )
    Timer1. Enabled = False
    SendMessage hwdc, WM_CAP_GET_FRAME, 0, 0
    SendMessage hwdc, WM_CAP_COPY, 0, 0
    Picture1. Picture = Clipboard. GetData
    GetBitmapBits Picture1. Picture, Pinfo. bmWidthBytes * Pinfo. bmHeight, _
      Pbytes(0, 0, 0)
    Gray Picture1. ScaleWidth, Picture1. ScaleHeight
    If blkflag = True Then Negative Picture1. ScaleWidth, Picture1. ScaleHeight
    Lowpass Picture1. ScaleWidth, Picture1. ScaleHeight
    Contrast Picture1. ScaleWidth, Picture1. ScaleHeight
    For i = 1 To Val(Text4. Text)
      Geometricmean Picture1. ScaleWidth, Picture1. ScaleHeight, gms
    Next i
    AdaptiveThreshold Picture1. ScaleWidth, Val(Text2. Text), Val(Text3. Text)
    Clean Picture1. ScaleWidth, Picture1. ScaleHeight, Val(Text3. Text)
    WhiteLineWidth Picture1. ScaleWidth, Val(Text2. Text), Val(Text3. Text)
    SetBitmapBits Picture2. Image. Pintb. bmWidthBytes * Pinfo. bmHeight, _
      Pbytes(0, 0, 0)
```

```
Picture2. Refresh
Picture2. Picture = Picture2. Image
 Picture2. Line ( Val ( Text2. Text ) - 2, Val ( Text3. Text ) - 2 ) - ( Val
(Text2. Text) _
    +2. Val(Text3. Text) +2),RGB(255,0,0),B

If PixelCountLeft < MIN_PATH_WIDT H And PixelCountRight < _
   MIN_PATH_WIDTH Then
   Text5. Text = " No path "
Else
   Text5. Text = " Path found "
End If
If PixelCountLeft > = MIN_PATH_WIDTH And PixelCountRight < _
   MIN_PATH_WIDTH Then
   Text6. Text =0:Text7. Text =50          '左转
   Text8. Text = " Turn left "
   Picture3. Picture = LoadPicture(" turn_left. jpg ")
End If
If PixelCountLeft < MIN_PATH_WIDTH And PixelCountRight > = _
   MIN_PATH_WIDTH Then
   Text6. Text =50:Text7. Text =0          '右转
   Text8. Text = " Turn right "
   Picture3. Picture = LoadPicture(" turn_right. jpg ")
End If
If PixelCountLeft > = MIN_PATH_WIDTH And PixelCountRight > = _
   MIN_PATH_WIDTH Then
   Text6. Text =100:Text7. Text =100      '向前
   Text8. Text = " Forward "
   Picture3. Picture = LoadPicture(" forward. jpg ")
End If
If PixelCountLeft < MIN_PATH_WIDTH And PixelCountRight < _
   MIN_PATH_WIDTH Then
   Text6. Text =0:Text7. Text =0           '无路径,忽略
   Text8. Text = ""
   Picture3. Picture = LoadPicture(" blank. jpg ")
```

```
        End If
        Timer1. Enabled = True
    End Sub
```

图 5. 15 所示图像文件用于表示导航方向。



图 5. 15 图像文件

如果采集图像的大小与图像框不符，那么将激活‘Format’菜单使得图像大小转换为 160 × 120 像素。如果没有可用的网络摄像头，那么会在消息框中弹出一条“未找到任何网络摄像头！（No webcam found!）”的消息。

5. 11 示例 10

目标：开发一个 VB6 程序用于采集和处理网络视频流，在基于视觉的导航中选择黑色/白色路径。根据路径是否存在来进行推理，应用合适的规则来确定不同的导航方向以及差动驱动的速度参数，同时添加声音输出以引起关注。

以下步骤对程序开发过程进行概述：

1）所有必需的 API 调用均在‘Webcam10. bas’模块中声明。同时在 VB6 程序的‘Form1’中必须包含该模块。

2）AVICAP32. DLL 用于通过调用合适的 API 来采集网络视频流。网络摄像头视频格式应为 RGB24 或 YUY2。

3）在 FORM1 中添加两个‘Picture Box’控件，‘Picture1’用于以 10ms 的固定间隔从视频流中采集图像，并将其作为剪贴板数据，‘Picture2’在‘Timer1’控件下以同样的速率对所采集的图像进行处理。

4）在‘Form1’中添加菜单项‘Format’，用于设置图像大小为 160 × 120 像素。

5）通过调用‘GetObject’API，从‘Picture1’中获得图像像素数据信息。

6）在‘Timer1’控件下通过调用‘GetBitmapBits’API 获得一个 8bit 的像素数组‘Pbytes（c，x，y）’。‘Pbytes’中的每个元素均包含图像坐标‘x’和‘y’处每个像素的 8bit RGB 像素信息。‘c’代表颜色；c:2 代表红色；c:1 代表绿色；c:0 代表蓝色。

7）点击‘cmdWhiteBlack’按钮，在‘Shape1’中显示路径颜色（白色或黑色）。

8）根据示例9处理采集图像。如果找到路径，则根据三条规则确定合适的导航方向（‘forward’或‘turn-left’或‘turn-right’）和相应的差动驱动速度参数。图像框显示导航方向。

9）调用程序‘SndPlaySound’以及合适的‘wave’文件，来激活声音输出。

下面给出‘Webcam10. bas’模块的程序：

```
Global Const WS_CHILD As Long = &H40000000
Global Const WS_VISIBLE AS Long = &H10000000
Global Const WM_USER = 1024
Global Const WM_CAP_DRIVER_CONNECT = WM_USER + 10
G10bal Const WM_CAP_SET_PREVIEW = WM_USER + 50
Global Const WM_CAP_SET_PREVIEWRATE = WM_USER + 52
Global Const WM_CAP_DRIVER_DISCONNECT As Long = WM_USER + 11
Global Const WM_CAP_DLG_VIDEOFORMAT As Long = WM_USER + 41
Global Const WM_CAP_DLG_VIDEOCOMPRESSION As Long = _
  WM USER + 46
Global Const WM_CAP_DLG_VIDEODISPLAY As Long = WM_USER + 43
Global Const WM_CAP_DLG_VIDEOSOURCE As Long = WM_USER + 42
Global Const WM_CAP_GET_FRAME As Long = 1084
Global Const WM_CAP_COPY As Long = 1054
Global Const WM_CAP_SET_SCALE As Integer = WM_USER + 53
Global Const SWP_NOMOVE As Integer = 2
Global Const SWP_NOZORDER As Integer = 4
Global Const HWND_BOTTOM As Integer = 1
Global Const SND_ASYNC = 1
Global Const SND_LOOP = 8
Global Const SND_NODEFAULT = 2
Global Const SND_SYNC = 0
Global Const SND_NOSTOP = 16
Global Const SND_MEMORY = 4

Declare Function SendMessage Lib " user32 " Alias " SendMessageA " ( ByVal
hwnd_
     As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As
Long)_
     As Long Declare Function capCreateCaptureWindow Lib " avicap32. dll "
```

```
Alias_
    " capCreateCaptureWindowA "( ByVal nWindowName As String, ByVal nStyle_
     As Long, ByVal nx As Integer, ByVal ny As Integer, ByVal nWidth As
Integer,_
    ByVal nHeight As Integer. ByVal nHwnd As Long, ByVal nld As Integer) As
Long
  Declare Function SetWindowPos Lib " user32 "( ByVal hwnd As Long,_
    ByVal hWndlnsertAfler As Long, ByVal x As Long, ByVal y As Long,_
    ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As Long
  Declare Function GetObject Lib " gdi32 " Alias " GetObjcctA "( ByVal hObject As
Long,_
    ByVal nCount As Long, lpObject As Any) As Long
  Declare Function GetBitmapBits Lib " gdi32 "( ByVal hBitmap As Long, ByVal dw-
Count_
    As Long, lpBits As Any) As Long
  Declare Function SetBitmapBits Lib " gdi32 "( ByVal hBitmap As Long, ByVal dw-
Count_
    As Long, lpBits As Any) As Long
  Declare Function sndPlaySound Lib " winmm. dll " Alias " sndPlaySoundA "_
    ( ByVal lpszSoundName As String, ByVal uFlags As Long) As Long
```

图 5. 16 给出了 ‘Form1’ 的布局。

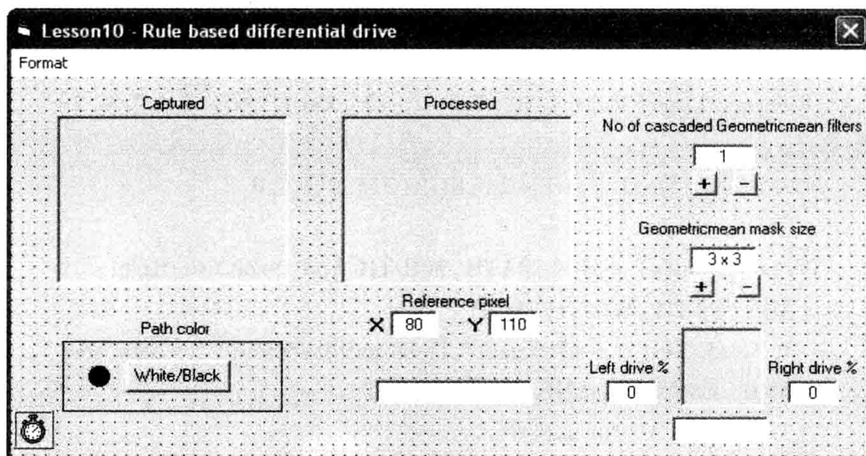


图 5. 16　‘Form1’ 的布局

接下来的文本显示了 ‘Timer1’ 的程序代码，其余代码请参见示例 9：

```
Private Sub Timer1_Timer( )
    Timer1. Enabled = False
    SendMessage hwdc, WM_CAP_GET_FRAME, 0, 0
    SendMessage hwdc, WM_CAP_COPY, 0, 0
    Picture1. Picture = Clipboard. GetData
    GetBitmapBits Picture1. Picture, Pinfo. bmWidthBytes * Pinfo. bmHeight, _
      Pbytes(0, 0, 0)

    Gray Picture1. ScaleWidth, Picture1. ScaleHeight
    If blkflag = True Then Negative Picture1. ScaleWidth, Picture1. ScaleHeight
    Lowpass Picture1. ScaleWidth, Picture1. ScaleHeight
    Contrast Picture1. ScaleWidth, Picture1. ScaleHeight
    For i = 1 To Val(Text4. Text)
      Geometricmean Picture1. ScaleWidth, Picture1. ScaleHeight. gms
    Next i
    AdaptiveThreshold Picture1. ScaleWidth, Val(Text2. Text), Val(Text3. Text)
    Clean Picture1. ScaleWidth, Picture1. ScaleHeight, Val(Text3. Text)
    WhiteLineWidth Picture1. ScaleWidth, Val(Text2. Text), Val(Text3. Text)

    SetBitmapBits Picture2. Image. Pinto. bmWidthBytes * Pinfo. bmHeight, _
      Pbytes(0, 0, 0)
    Picture2. Refresh
    Picture2. Picture = Picture2. Image
     Picture2. Line ( Val ( Text2. Text ) - 2, Val ( Text3. Text ) - 2 ) - ( Val
(Text2. Text)_
       + 2. Val(Text3. Text) + 2), RGB(255, 0, 0), B

    If PixelCountLeft < MIN_PATH_WIDTH And PixelCOuntRight < _
      MIN_PATH_WIDTH Then
      If Text5. Text < > " No path " Then sndPlaySound " No path. wav ", _
      SND_ASYNC Or SND_NODEFAULT
      Text5. Text = " No path "
    Else
      If Text5. Text < > " Path found " Then sndPlaySound " Path found. wav ", _
        SND_ASYNC Or SND_NODEFAULT
```

```
        Text5. Text = " Path found "
    End If

    If PixelCountLeft > = MIN_PATH_WIDTH And PixelCountRight < _
        MIN_PATH_WIDTH Then
        Text6. Text = 0 : Text7. Text = 50   '左转
        Text8. Text = " Turn left "
        Picture3. Picture = LoadPicture( " turn_left. jpg " )
    End If
    If PixelCountLeft < MIN_PATH_WIDTH And PixelCountRight > = _
        MIN_PATH_WIDTH Then
        Text6. Text = 50 : Text7. Text = 0   '右转
        Text8. Text = " Turn right "
        Picture3. Picture = LoadPicture( " turn_right. jpg " )
    End If
    If PixelCountLeft > = MIN_PATH_WIDTH And PixelCountRight > = _
        MIN_PATH_WIDTH Then
        Text6. Text = 100 : Text7. Text = 100   '向前
        Text8. Text : " Forward "
        Picture3. Picture = LoadPicture( " forward. jpg " )
    End If
    If PixelCountLeft < MIN_PATH_WIDTH And PixelCountRight < _
        MIN_PATH_WIDTH Then
        Text6. Text = 0 : Text7. Text = 0   '无路径,忽略
        Text8. Text = ""
        Picture3. Picture = LoadPicture( " blank. jpg " )
    End If
    Timer1. Enabled = True
End Sub
```

必要时，通过 PC 的声卡来播放两个预先录制的声音文件 ‘Nopath. wav’ 和 ‘Pathfound. wav’。PC 的声音录制程序可用来创建这些声音文件。

如果采集图像的大小与图像框不符，那么将激活 ‘Format’ 菜单使得图像大小转换为 160 × 120 像素。如果没有可用的网络摄像头，那么会在消息框中弹出一条“未找到任何网络摄像头！(No webcam found!)”的消息。

5.12 本章小结

在此，一步步地介绍了十个示例来展示在32位Windows环境下，视觉导航应用程序的开发技巧。

示例1：该示例演示了如何从低成本网络摄像头的视频流中采集图像帧，并通过鼠标查看像素值。

示例2：该示例演示了如何处理来自视频流的采集图像帧，包括彩色—灰度图像变换和低通滤波两种处理。

示例3：增加基于直方图拉伸技术的对比度增强方法，来提高光照较差条件下的对比度。

示例4：增加几何均值滤波器来平滑和抑制图像细节。

示例5：在光照变化条件下，引入一个自适应阈值来提取白色路径。

示例6：提供一个清除操作来去除检测到的不必要物体。

示例7：增加一种路径颜色选择（白色或黑色）的操作。

示例8：以一个固定像素点为参考，调整导航中白色或黑色路径的搜索。

示例9：引入一种基于规则的方法来确定导航中一个差动驱动系统的左、右轮速度设定。

示例10：在导航过程中，增加声音输出以引起注意。

参考文献

[1] Balena, F.: Programming Microsoft Visual Basic 6. Microsoft Press (1999)
[2] Mandelbrot Set International Ltd., Advanced Microsoft Visual Basic 6. Microsoft Press (1998)
[3] Appleman, D.: Dan Appleman's Win32 API Puzzle Book and Tutorial for Visual Basic Programmers. Apress (1999)
[4] Gonzalez, Woods: Digital Image Processing. Prentice Hall (2002)

第6章　基于视觉的移动机器人路径跟踪

摘要： 本章讨论了如何开发一种基于视觉的导航方法来跟踪室内路径，使得机器人能够跟踪一条细线或沿着一条较宽的路径运动。在此，该方法中采用模糊逻辑来达到预期目标。在前方没有障碍物的情况下，该方法将基于模糊视觉的导航方法来对机器人进行导航。另外，当检测到有任何障碍物时，该方法还采用了一种基于红外的模糊避障策略。

6.1　简介

本章将介绍一种基于视觉的导航算法，并结合实验室自主开发的机器人，在存在或没有障碍的情况下，采用模糊逻辑来进行路径跟踪[10]。近年来，在移动机器人导航中，模糊逻辑作为一种可行方法已被广泛接受。在参考文献［1］中，首次开发了一种用于避障的模糊控制器。其中，控制器采用一种基于视觉的算法从摄像机中获取机器人前方障碍区域和空白区域的相关信息，并利用仿真器来推导模糊规则。在参考文献［2］中的另一项关于走廊导航的类似研究工作是在所开发的 ATHENE 智能车上采用了一个基于视频图像的模糊控制器。十多年来，模糊逻辑方法广泛用于移动机器人的导航系统。参考文献［3］中提出了一种用于移动机器人控制的基于模糊逻辑的路径规划和执行方法。几乎同一时期，参考文献［4］中提出的一种用于移动机器人视觉导航的模糊逻辑方法，在防撞和避障方面得到了成功应用。在参考文献［5］中，提出了一种基于预测学习模糊控制的新方法，其中通过一个 CCD 摄像头来获取环境信息。在该研究工作中，所采集的图像被分为几种特征模板，对于每种模板，根据控制规则集对机器人进行编程，并设为先验知识。通过模板匹配，机器人将这些规则集整合成单个集合。另外，还有一些研究工作是利用模糊控制器并结合立体视觉系统和一些传统传感器来进行机器人导航[6,7]。在参考文献［8］中，对几种基于模糊逻辑的移动机器人导航方法进行了详细而全面的介绍。近年来，一种称为基于规则的模糊通行性指数方法用于移动机器人的室外导航，其中通常根据图像数据进行不同特征分类，如地形粗糙度、地形坡度、地形不连续性以及地形硬度等特征[9]。一旦从可视场景中提取出这些特征，则可制订用于移动机器人平滑导航的通行性指数模糊规则。

在此，采用第4章所介绍的本实验室自主开发的机器人，并提出一种基于

模糊逻辑的移动机器人导航新方法。在存在障碍或没有障碍的情况下，该方法能够跟踪窄路径或宽路径的中线。机器人前方没有障碍时，该系统采用一个基于视觉的模糊导航模块。一旦机器人感知到前方有障碍物，则停用该模糊模块，并激活一个基于红外的模糊避障模块，以使得机器人能够安全避障，并重新定位在路径的中线上。如果完成该目标，则停用基于红外的模糊模块，并重新激活基于视觉的模糊模块，使得机器人继续跟踪路径。机器人系统中采用了第4章中所述的适合于客户端/服务器操作的内网连接功能，从而可正确选择机器人的功能，并能够在远端客户端的PC对机器人发出合适的控制指令。

6.2 所提方法概述

图6.1给出了本书研究工作所开发的完整方案。设在当前给定时刻，差动驱动机器人系统的位置为（x_R，y_R，ϕ_R）。根据机器人前方的环境，对机器人发出一个新的导航命令，该命令包括线速度指令(v)和转向角指令(θ)。转向角指

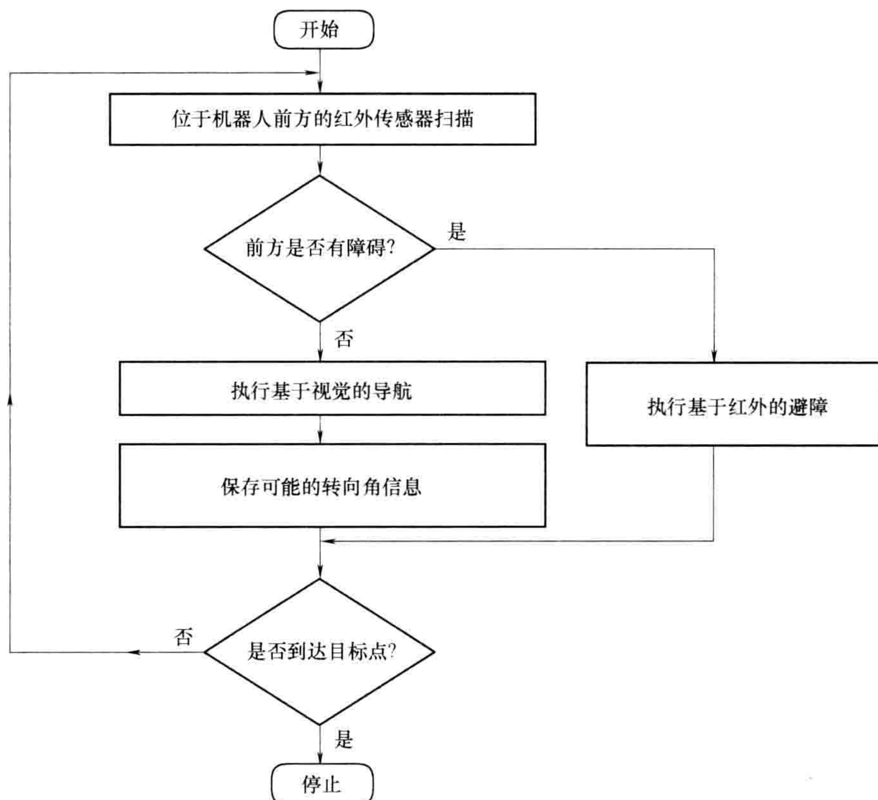


图6.1　移动机器人的导航策略

令可以是相对于机器人当前位置以逆时针方向感知的(0° ~ 180°) 范围内的任意值。图 6.2 给出了由 *XWY* 表示的世界坐标系以及由 *xoy* 表示的移动机器人坐标系，其中 *o* 为机器人的中心。新的机器人导航方向是沿图 6.2 中 *op* 的方向。在任意给定位置，机器人利用位于 4、5 和 6 点处的红外传感器扫描前方，以确定前方区域是否有障碍物。如果检测到障碍物，机器人将首先发出一条前方有障碍的语音消息，希望在听到机器人语音消息后，可将误放在机器人路径上的障碍物清除。如果障碍物没有清除，机器人将根据在 11 个扫描位置得到的红外数据进行避障。避障策略是机器人应暂时偏离其原路径，成功避障后，仍然返回到理想的原路径。一旦机器人重新检测到前方没有障碍，系统将返回到基于视觉的导航控制机制。所开发的本系统在基于视觉的导航和基于红外的避障中都采用一种模糊导航算法。导航机制的基本原理是无论路径上是否具有障碍物，机器人都应跟踪朝向目标点的一条路径的中线。对于较宽的路径，机器人总是沿路径中线进行导航。同理，对于较窄的路径，即路径宽度小于机器人本体宽度，导航算法也是试图跟踪该路径的中线。

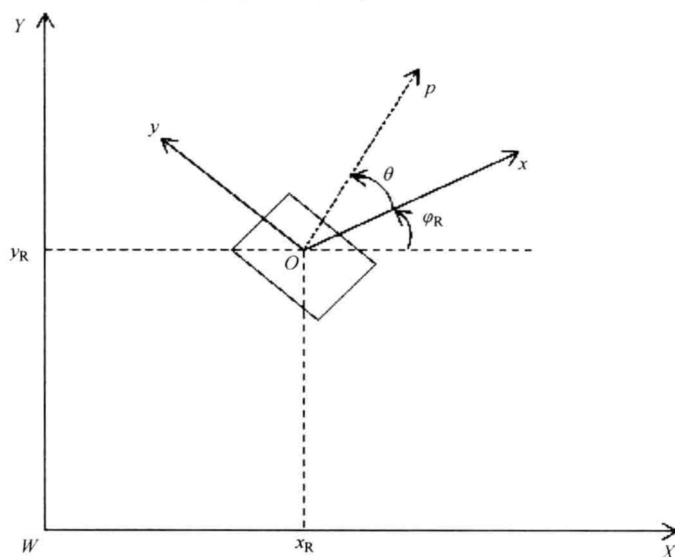


图 6.2　机器人的坐标系

6.3　基于视觉的机器人模糊导航系统

如果机器人发现前方没有障碍，将执行基于视觉的导航。该导航机制采用了如下图像处理步骤：

1. 从网络摄像头的视频流中采集一帧图像

当经过红外系统扫描推断出机器人前方区域无障碍时，在机器人上笔记本电脑网络摄像头所获取的连续视频流中采集一帧图像，该帧图像中包含了机器人前方环境的可视化信息。对该帧图像进一步处理从中提取有用信息，首先将彩色图像转换为相应的灰度图像，并对该灰度图像执行图像处理步骤。

2. 在环境的灰度图像中提取路径

接下来是对图像进行分割，使得能够从环境背景中提取出路径。为此，首先对该图像进行低通滤波来去除噪点，然后自动校正图像亮度，如果由于光线较暗而造成图像较黑，则通过改变亮度值的动态范围来增强图像的整体亮度。然后对亮度校正后的图像进行处理，利用一种类似于结构元素膨胀算法的操作，来连接并加粗孤立的亮点。这种连接加粗操作可通过一种几何平均滤波技术来实现。

最后，对加粗后的图像进行阈值分割。为在环境背景中仅提取路径，在此选取较大的亮度阈值。图 6.3 和图 6.4 分别给出了经过上述各个图像处理步骤之后，没有进行几何均值滤波和经过几何均值滤波后的一个示例环境。图 6.5 和图 6.6 给出了类似的情况，只是在路径上有一个障碍物。图 6.6 给出图如何利用几何平均滤波去除图像分割后的干扰物体，并清晰地提取路径，这在图 6.5 中是无法实现的。

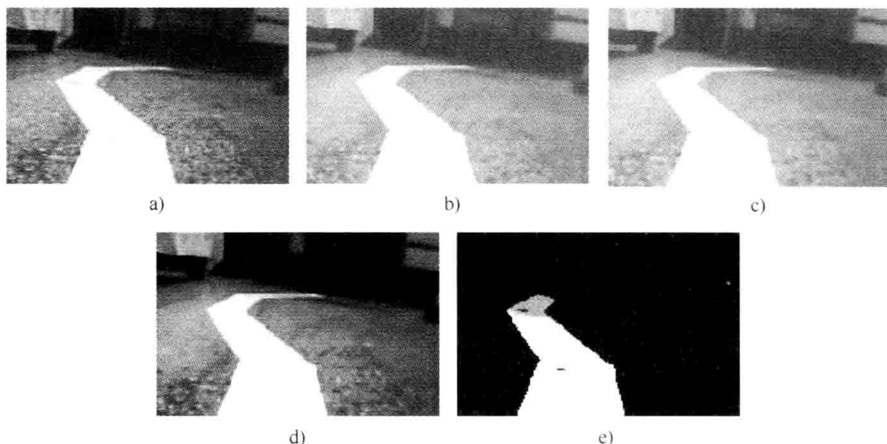


图 6.3 一个示例环境的图像处理结果

a）采集的原始图像 b）相应的灰度图像 c）去噪后的图像

d）亮度自动校正后的图像 e）阈值分割后的最终图像

3. 采用基于视觉的模糊导航系统

根据上一步骤中所获得的阈值图像，开发一个基于模糊逻辑的导航系统。在本例中，图像大小为 160 × 120 像素，其中左上角像素点作为坐标原点（0，0），右下角像素点对应坐标（159，119）。然后在图像的垂直中线上选择一个种

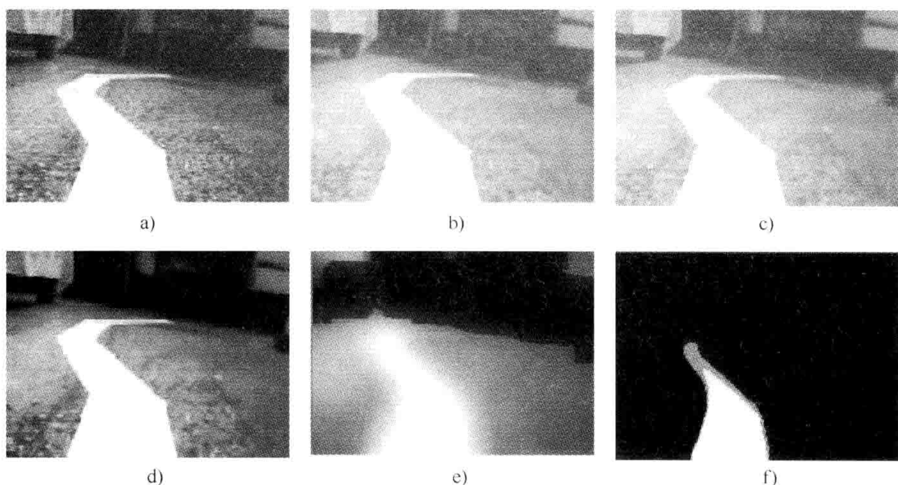


图 6.4　图 6.3 中示例环境的图像处理结果

a）采集的原始图像　b）相应的灰度图像　c）去噪后的图像　d）亮度自动校正后的图像　e）采用几何均值滤波进行孤立点连接与加粗后的图像　f）阈值分割后的最终图像

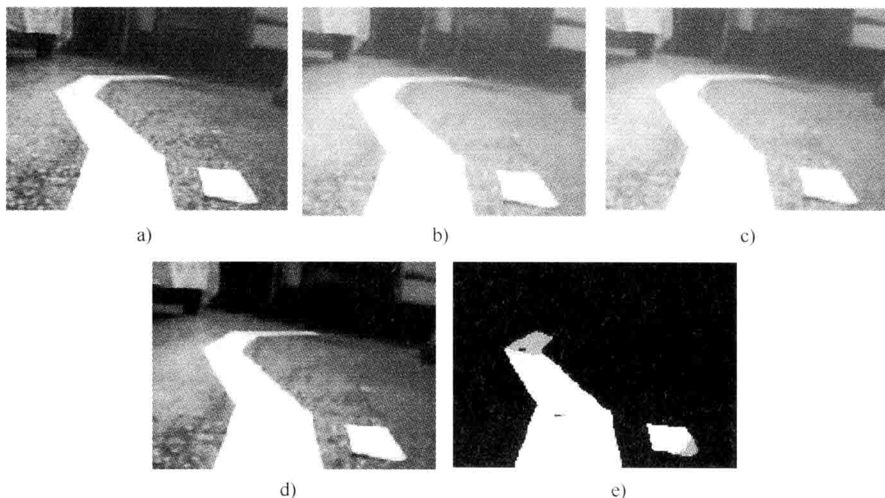


图 6.5　具有干扰物体的示例环境的图像处理结果

a）采集的原始图像　b）相应的灰度图像　c）去噪后的图像
d）亮度自动校正后的图像　e）阈值分割后的最终图像

子点 S，该种子点越朝向图像底部，表明该点实际位置对应于越靠近机器人的实际环境点。在图像坐标系中，该种子点选为（80，110）。在该点位置处在图像中绘制一条水平线。从该种子点 S 开始，可向左和向右来遍历该扫描线，由此计算这些具有亮度值的像素点个数（包括 S 点左侧和右侧的所有像素点），从而确定机器人左侧和右侧的路径宽度。如果两侧的像素点个数相同，即可推断机

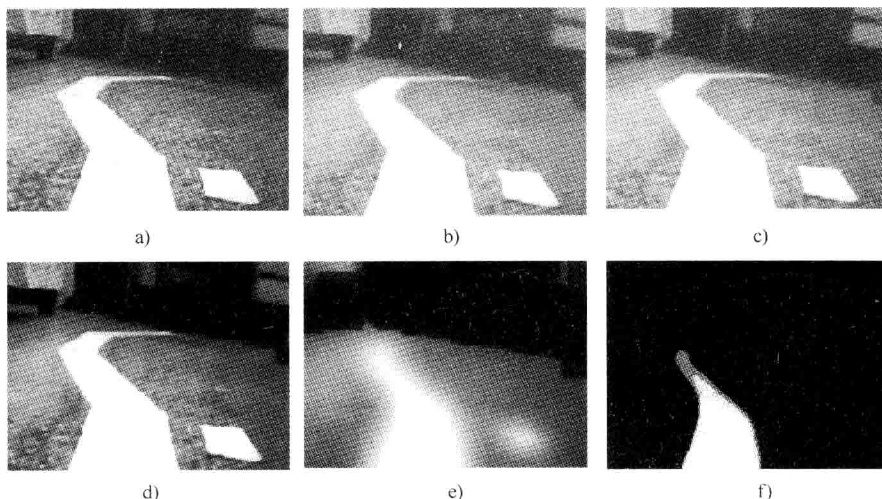


图 6.6 具有干扰物体的示例环境的图像处理结果

a）采集的原始图像 b）相应的灰度图像 c）去噪后的图像 d）亮度自动校正后的图像
e）采用几何均值滤波进行孤立点连接与加粗后的图像 f）阈值分割后的最终图像

器人大致位于路径的中线上。另一方面，如果左侧像素点个数大于右侧像素点个数，则表明机器人位置偏向路径的右侧，然后模糊推断系统就试图使机器人朝路径中线运动。如果右侧像素点个数大于左侧像素点个数，则表明机器人偏向路径左侧，然后模糊系统就给机器人发出相反命令，使得机器人向路径中线运动。该基于模糊逻辑的视觉导航系统是一个双输入双输出系统，其中两个输入变量为 PixelCountLeft 和 PixelCountRight，两个输出变量分别为线速度指令(v)和转向角指令(θ)。所开发的模糊系统是一种零阶 Takas-Sugano(TS)系统。为使得该模糊系统的鲁棒性更强，将种子点左侧和右侧的像素个数取为在三个种子点 $S_1\equiv(80,\ 109)$、$S_2\equiv(80,\ 110)$ 和 $S_3\equiv(80,\ 111)$ 处绘制的三条连续的水平线，然后计算其平均值为

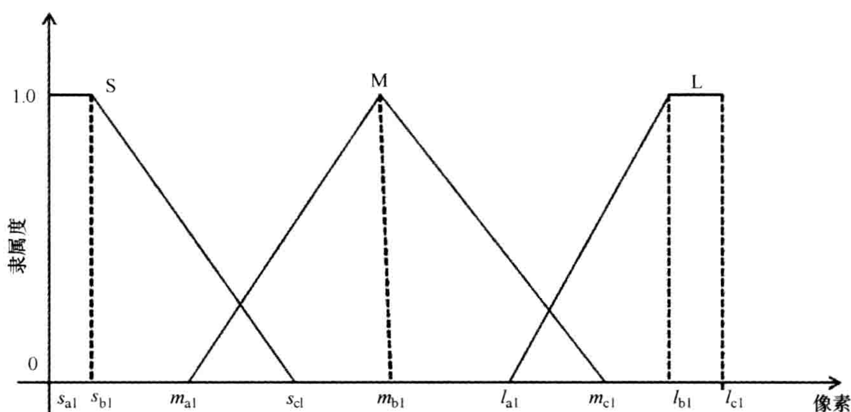
$$\text{PixelCountLeft} = \frac{1}{3}\sum_{i=1}^{3}\text{pcl}_i \tag{6.1a}$$

$$\text{PixelCountRight} = \frac{1}{3}\sum_{i=1}^{3}\text{pcr}_i \tag{6.1b}$$

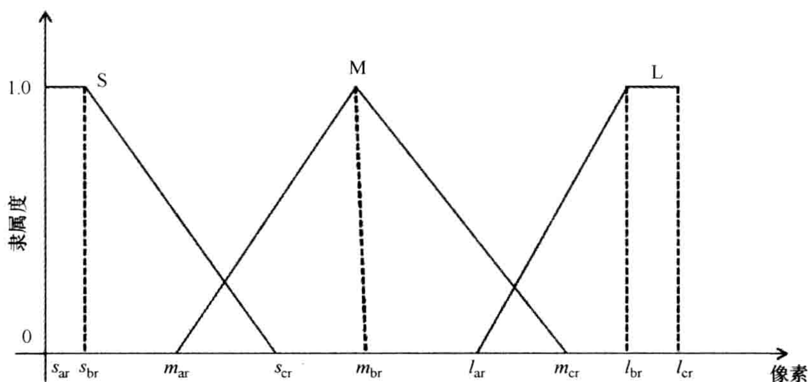
式中 pcl_i——种子点 S_i 沿扫描线向左的像素个数；

pcr_i——种子点 S_i 沿扫描线向右的像素个数。

图 6.7 给出了模糊化时的输入隶属函数，其中每个输入变量都用小(S)、中(M)和大(L)三种隶属函数模糊化。输入变量的模糊集或隶属函数分别表示为



a)



b)

图 6.7　PixelCountLeft(见图 6.7a)和 PixelCountRight(见图 6.7b)的隶属函数

$$\mu_s(\text{PixelCountLeft}) = \begin{cases} \left(\dfrac{s_{cl} - \text{PixelCountLeft}}{s_{cl} - s_{bl}}\right) & s_{bl} < \text{PixelCountLeft} < s_{cl} \\ 1 & 0 \leqslant \text{PixelCountLeft} \leqslant s_{bl} \\ 0 & \text{其他} \end{cases} \tag{6.2a}$$

$$\mu_M(\text{PixelCountLeft}) = \begin{cases} \left(\dfrac{\text{PixelCountLeft} - m_{al}}{m_{bl} - m_{al}}\right) & m_{al} < \text{PixelCountLeft} < m_{bl} \\ \left(\dfrac{m_{cl} - \text{PixelCountLeft}}{m_{cl} - m_{bl}}\right) & m_{bl} \leqslant \text{PixelCountLeft} \leqslant m_{cl} \\ 0 & \text{其他} \end{cases} \tag{6.2b}$$

$$\mu_{\mathrm{L}}(\text{PixelCountLeft})=\begin{cases}\left(\dfrac{\text{PixelCountLeft}-l_{\mathrm{al}}}{l_{\mathrm{bl}}-l_{\mathrm{al}}}\right) & l_{\mathrm{al}}<\text{PixelCountLeft}<l_{\mathrm{bl}}\\ 1 & \text{PixelCountLeft}\geqslant \boldsymbol{l}_{\mathrm{bl}}\\ 0 & \text{其他}\end{cases}\tag{6.2c}$$

$$\mu_{\mathrm{s}}(\text{PixelCountRight})=\begin{cases}\left(\dfrac{s_{\mathrm{cr}}-\text{PixelCountRight}}{s_{\mathrm{cr}}-s_{\mathrm{br}}}\right) & s_{\mathrm{br}}<\text{PixelCountRight}<s_{\mathrm{cr}}\\ 1 & 0\leqslant\text{PixelCountRight}\leqslant s_{\mathrm{br}}\\ 0 & \text{其他}\end{cases}\tag{6.3a}$$

$$\mu_{\mathrm{M}}(\text{PixelCountRight})=\begin{cases}\left(\dfrac{\text{PixelCountRight}-m_{\mathrm{ar}}}{m_{\mathrm{br}}-m_{\mathrm{ar}}}\right) & m_{\mathrm{ar}}<\text{PixelCountRight}<m_{\mathrm{br}}\\ \left(\dfrac{m_{\mathrm{cr}}-\text{PixelCountRight}}{m_{\mathrm{cr}}-m_{\mathrm{br}}}\right) & m_{\mathrm{br}}\leqslant\text{PixelCountRight}\leqslant m_{\mathrm{cr}}\\ 0 & \text{其他}\end{cases}\tag{6.3b}$$

$$\mu_{\mathrm{L}}(\text{PixelCountRight})=\begin{cases}\left(\dfrac{\text{PixelCountRight}-l_{\mathrm{ar}}}{l_{\mathrm{br}}-l_{\mathrm{ar}}}\right. & l_{\mathrm{ar}}<\text{PixelCountRight}<l_{\mathrm{br}}\\ 1 & \text{PixelCountRight}\geqslant l_{\mathrm{br}}\\ 0 & \text{其他}\end{cases}\tag{6.3c}$$

对于每个输出变量，输出由单变量表示。模糊规则库由一组 IF-THEN 模糊规则组成。一条模糊规则 i 可表示为

$\boldsymbol{R}_{\mathrm{vis}}^{(i)}$：IF　x_1 是 M_{i1} 和 x_2 是 M_{i2}

THEN y_1 是 V_{vis_i1} 和 y_2 是 θ_{vis_i2}，$i=1, 2, \cdots, N$　　(6.4)

式中　$\boldsymbol{x}=[x_1, x_2]^{\mathrm{T}}=[\text{PixelCountLeft}, \text{PixelCountRight}]^{\mathrm{T}}$；

$\boldsymbol{y}=[y_1, y_2]^{\mathrm{T}}=[v_{\mathrm{vis}}, \boldsymbol{\theta}_{\mathrm{vis}}]^{\mathrm{T}}$；

$$M_{i1}\in\{S,M,L\};$$

$$M_{i2}\in\{S,M,L\};$$

$$V_{\mathrm{vis}_i1}\in\boldsymbol{\nu}_{\mathrm{vis}};$$

$$\theta_{\mathrm{vis}_i2}\in\theta_{\mathrm{vis}};$$

$\boldsymbol{\nu}_{\mathrm{vis}}$ = 输出线速度单矢量 $=[\nu_{\mathrm{vis}1}, \nu_{\mathrm{vis}2}, \cdots, \nu_{\mathrm{vis}N}]^{\mathrm{T}}$；

$\boldsymbol{\theta}_{\mathrm{vis}}$ = 输出转向角单矢量 $=[\theta_{\mathrm{vis}1}, \theta_{\mathrm{vis}2}, \cdots, \theta_{\mathrm{vis}N}]^{\mathrm{T}}$；

N——模糊规则库中规则的总数。

表 6.1 视觉导航系统的模糊规则库

规则号	前件(IF 子句)		后件(THEN 子句)	
	PixelCountLeft	PixelCountRight	$\boldsymbol{\nu}_{vis}$(标幺值)	$\boldsymbol{\theta}_{vis}$/°
1	小	小	0.9	90
2	小	中	0.5	67
3	小	大	0.1	45
4	中	小	0.5	112
5	中	中	0.9	90
6	中	大	0.5	67
7	大	小	0.1	135
8	大	中	0.5	112
9	大	大	0.9	90

所构建的模糊规则库见表 6.1。线速度的模糊输出为一个标幺值，该值是乘以一个合适的增益($\boldsymbol{K}_{vel_vis}$)，通过采用加权平均法来去模糊化，然后输出线速度指令($\boldsymbol{\nu}_{vis}$)和输出转向角指令($\boldsymbol{\theta}_{vis}$)计算如下：

$$\boldsymbol{\nu}_{vis} = (\boldsymbol{K}_{vel_vis}) * \left(\frac{\sum_{i=1}^{N} \boldsymbol{\nu}_{visi} * \boldsymbol{\alpha}_i(x)}{\sum_{i=1}^{N} \boldsymbol{\alpha}_i(x)} \right) \tag{6.5}$$

$$\boldsymbol{\theta}_{vis} = \frac{\sum_{i=1}^{N} \boldsymbol{\theta}_{visi} * \boldsymbol{\alpha}_i(x)}{\sum_{i=1}^{N} \boldsymbol{\alpha}_i(x)} \tag{6.6}$$

式中 $\boldsymbol{\alpha}_i(x)$——规则 $i = \prod_{j=1}^{2} \boldsymbol{\mu}_i(x_j)$ 的触发强度。

4. 如果机器人偏离路径则保存可能的绕行转向角

机器人导航系统中配置了一个附加模块来应对紧急情况。考虑到在某些环境下，机器人偏离了路径，并且在处理后的输出图像中，变量 PixelCountLeft 和 PixelCountRight 均计算为零。在这种情况下，以迭代方式给定机器人一个很小的转向角绕行命令(线速度取为零)，直到在变量 PixelCountLeft 和 PixelCountRight 中，至少有一个不为零。由此可推断机器人已朝向原路径，因此随后激活的基于视觉的导航算法将试图使机器人返回到路径的中线。现在，机器人应按顺时针方向还是按逆时针方向绕行取决于机器人在之前的迭代过程中是朝左运动还是朝右运动。这有助于减少绕行时所花费的时间，并能够恢复导航的原方向。

图 6.8 给出了一种保存可能转向角绕行信息的算法。在每个采样时刻(k)，

通过一次水平扫描来计算种子点左侧像素个数(pcl_k)和右侧像素个数(pcr_k),从而确定亮点的个数。为保证机器人转向正确,该过程要重复执行(N_rows)次来确定在采样时刻 k 的累积值 cum_pcl_k 和 cum_pcr_k。在基于视觉导航的每帧处理图像中重复执行该过程,以确定在采样时刻 k 所保存的两个相应量的最后值。然而在保存这些值时,当前采样时刻的优先级最高,随着返回过去时刻,相应的优先级也逐步减小。这可由下式表示:

$$store_pcl_k = k_1 \times cum_pcl_k + k_2 \times store_pcl_{k-1} \tag{6.7}$$

$$store_pcr_k = k_1 \times cum_pcr_k + k_2 \times store_pcr_{k-1} \tag{6.8}$$

在该系统中,遗忘因子取为 $k_1 = 0.25$ 和 $k_2 = 0.75$。若基于视觉的导航算法正常运行,则继续保存。然而只有当由于某些原因,机器人偏离路径且变量 PixelCountLeft 和 PixelCountRight 为零时,这些保存值才开始有用。接下来,根据($store_pcl_k_store_pcr_k$)的极性,选择有效的转向绕行方向。

如果值为正,则选择逆时针绕行方向,否则选择顺时针绕行方向。一旦绕行方向固定,就执行迭代过程。在此,机器人的线速度为零,固定旋转 10°。采集一帧图像,并按之前所述的图像处理步骤进行处理,来确定变量 PixelCountLeft 和 PixelCountRight 的值。如果至少一个值不为零,则重新激活基于视觉的导航算法,否则机器人旋转 10°并执行后续步骤。迭代执行上述处理过程,直到重新激活基于视觉的导航算法。

6.4 基于红外传感器的模糊避障算法

如果系统检测到前方有障碍,则激活基于红外的避障模块,并停用视觉系统。一个基于模糊逻辑的红外避障机制将使得机器人绕开障碍并继续沿原路径运动。一旦机器人避开障碍,则重新激活基于视觉的导航算法,这将使得机器人自动返回到路径的中线。对于所开发的红外模糊系统,每个红外传感器在 11 个角度($l = 1, 2, \cdots, 11$)进行扫描,从而产生 11 个红外读数 $IR_Sensor_val(l)$,单位为伏(V)。这 11 个读数分为三组 ,即 $IR_Group_val(p)$,$p = 1, 2, 3$。这样做是为了减少模糊系统的输入维数。在每个红外传感器组中,选取最大扫描读数作为该组的读数。因为读数越高,表明障碍越近。因此,这些模拟组读数可表示为

$$IR_Group_val(1) = \max(IR_Sensor_val(l) \mid l = 1,2,3,4) \tag{6.9}$$

$$IR_Group_val(2) = \max(IR_Sensor_val(l) \mid l = 5,6,7) \tag{6.10}$$

$$IR_Group_val(3) = \max(IR_Sensor_val(l) \mid l = 8,9,10,11) \tag{6.11}$$

接下来,开发一个三输入两输出的模糊避障系统,其中三个输入为 $IR_Group_val(p)$,$p = 1, 2, 3$,两个输出为(ν, θ)。在此还开发了一个零阶 TS 模

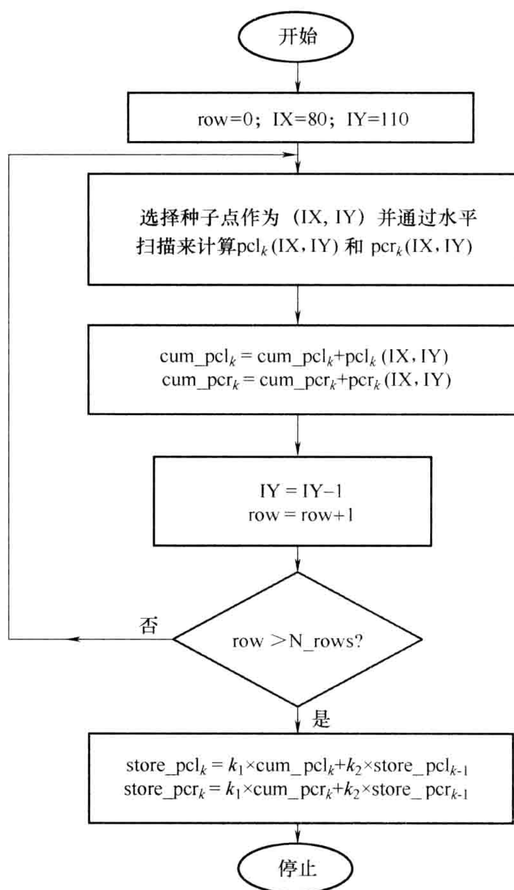
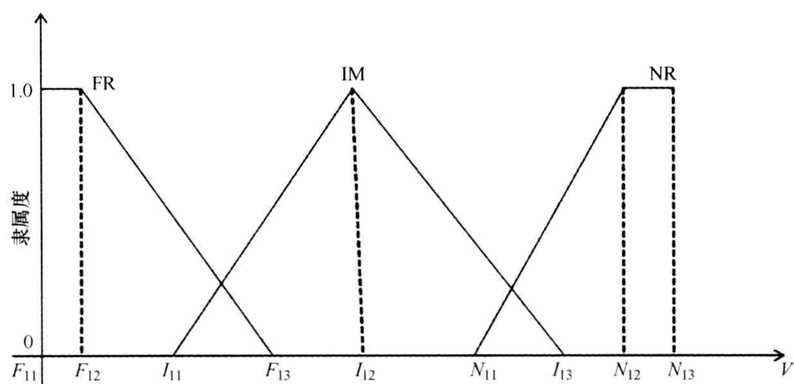


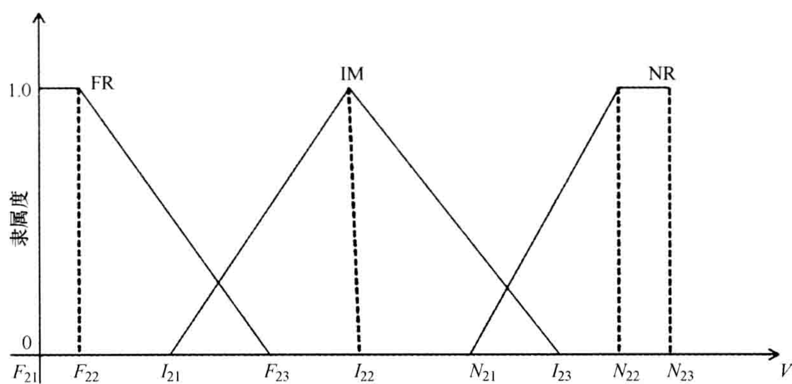
图 6.8　可能转向角绕行模块的信息存储过程

糊系统。图 6.9 给出了对每个输入变量所选择的隶属函数，每个输入采用三个隶属函数，即远(FR)、中(IM)和近(NR)来模糊化。相应的隶属函数可表示为

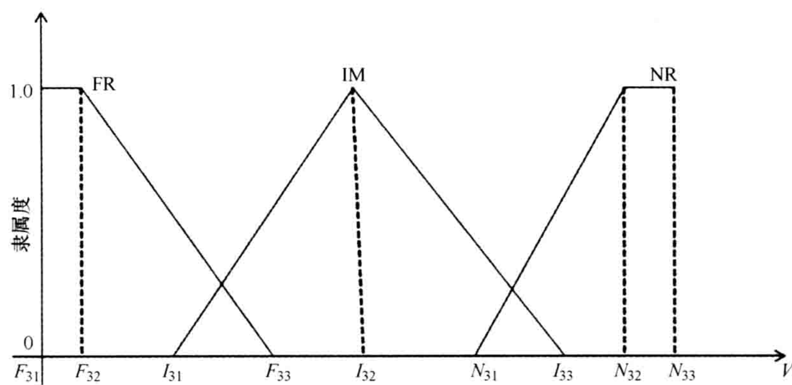
$$\mu_{\mathrm{FR}}(\mathrm{IR_Group_val}(p))=\begin{cases}\left(\dfrac{F_{p3}-\mathrm{IR_Group_val}(p)}{F_{p3}-F_{p2}}\right) & F_{p2}<\mathrm{IR_Group_val}(p)<F_{p3}\\ 1 & 0\leqslant \mathrm{IR_Group_val}(p)\leqslant F_{p2}\\ 0 & \text{其他}\end{cases}\tag{6.12a}$$



a)



b)



c)

图 6.9 隶属函数

a) IR_Group_val(1) b) IR_Group_val(2) c) IR_Group_val(3)

$$\mu_{\mathrm{IM}}(\mathrm{IR_Group_val}(p))=\begin{cases}\left(\dfrac{\mathrm{IR_Group_val}(p)-I_{p1}}{I_{p2}-I_{p1}}\right) & I_{p1}<\mathrm{IR_Group_val}(p)<I_{p2}\\ \left(\dfrac{I_{p3}-\mathrm{IR_Group_val}(p)}{I_{p3}-I_{p2}}\right) & I_{p2}\leqslant\mathrm{IR_Group_val}(p)<I_{p3}\\ 0 & \text{其他}\end{cases} \tag{6.12b}$$

$$\mu_{\mathrm{NR}}(\mathrm{IR_Group_val}(p))=\begin{cases}\left(\dfrac{\mathrm{IR_Group_val}(p)-N_{p1}}{N_{p2}-N_{p1}}\right) & N_{p1}<\mathrm{IR_Group_val}(p)<N_{p2}\\ 1 & \mathrm{IR_Group_val}(p)\geqslant N_{p2}\\ 0 & \text{其他}\end{cases} \tag{6.12c}$$

模糊规则 i 可表示为

$R_{\mathrm{obs}}^{(i)}$: IF　z_1 是 Q_{i1} 和 z_2 是 Q_{i2} 和 z_3 是 Q_{i3}

THEN u_1 是 $V_{\mathrm{obs_}i1}$ 和 u_2 是 $\theta_{\mathrm{obs_}i2}$，$i=1,2,\cdots,L$　(6.13)

式中　$z=[z_1, z_2, z_3]^{\mathrm{T}}=[\mathrm{IR_Group_val}(1), \mathrm{IR_Group_val}(2), \mathrm{IR_Group_val}(3)]^{\mathrm{T}}$;

$$\boldsymbol{u}=[u_1, u_2]^{\mathrm{T}}=[\boldsymbol{\nu}_{\mathrm{obs}}, \theta_{\mathrm{obs}}]^{\mathrm{T}};$$

$$Q_{i1}\in\{\mathrm{FR},\mathrm{IM},\mathrm{NR}\};$$

$$Q_{i2}\in\{\mathrm{FR},\mathrm{IM},\mathrm{NR}\};$$

$$Q_{i3}\in\{\mathrm{FR},\mathrm{IM},\mathrm{NR}\};$$

$$V_{\mathrm{obs_}i1}\in\boldsymbol{\nu}_{\mathrm{obs}};$$

$$\boldsymbol{\theta}_{\mathrm{obs_}i2}\in\boldsymbol{\theta}_{\mathrm{obs}};$$

$$\boldsymbol{\nu}_{\mathrm{obs}}=\text{输出线速度单矢量}=[\nu_{\mathrm{obs}1},\nu_{\mathrm{obs}2},\cdots,\nu_{\mathrm{obs}L}]^{\mathrm{T}};$$

$$\boldsymbol{\theta}_{\mathrm{obs}}=\text{输出转向角单矢量}=[\theta_{\mathrm{obs}1},\theta_{\mathrm{obs}2},\cdots,\theta_{\mathrm{obs}L}]^{\mathrm{T}};$$

L——该模糊系统的规则总数。

表 6.2 给出了避障时的整个模糊规则库。设 $K_{\mathrm{vel_obs}}$ 为线速度的比例增益，输出线速度指令($\boldsymbol{\nu}_{\mathrm{obs}}$)和输出转向角指令($\boldsymbol{\theta}_{\mathrm{obs}}$)可用下式计算：

$$\boldsymbol{\nu}_{\mathrm{obs}}=(K_{\mathrm{vel_obs}})*\left(\frac{\sum_{i=1}^{L}\boldsymbol{\nu}_{\mathrm{obs}i}*\boldsymbol{\beta}_i(z)}{\sum_{i=1}^{L}\boldsymbol{\beta}_i(z)}\right) \tag{6.14}$$

$$\boldsymbol{\theta}_{\text{obs}} = \frac{\sum_{i=1}^{L} \boldsymbol{\theta}_{\text{obs}i} * \boldsymbol{\beta}_i(z)}{\sum_{i=1}^{L} \boldsymbol{\beta}_i(z)} \tag{6.15}$$

式中 $\boldsymbol{\beta}_i(z)$——规则 $i = \prod_{j=1}^{3} \mu_i(z_j)$ 的触发强度。

表 6.2 避障时的模糊规则库

规则号	前件(IF 子句)			后件(THEN 子句)	
	IR_Gourp_val(1)	IR_Gourp_val(2)	IR_Gourp_val(3)	$\boldsymbol{\nu}_{\text{obs}}$(标幺值)	$\boldsymbol{\theta}_{\text{obs}}$/(°)
1	FR	FR	FR	0.8	0
2	FR	FR	IM	0.8	90
3	FR	FR	NR	0.7	90
4	FR	IM	FR	0.5	135
5	FR	IM	IM	0.7	135
6	FR	IM	NR	0.6	150
7	FR	NR	FR	0.3	135
8	FR	NR	IM	0.5	135
9	FR	NR	NR	0.4	160
10	IM	FR	FR	0.9	90
11	IM	FR	IM	0.8	90
12	IM	FR	NR	0.6	90
13	IM	IM	FR	0.7	50
14	IM	IM	IM	0.3	90
15	IM	IM	NR	0.2	105
16	IM	NR	FR	0.5	35
17	IM	NR	IM	0.2	105
18	IM	NR	NR	0.1	150
19	NR	FR	FR	0.8	90
20	NR	FR	IM	0.7	90
21	NR	FR	NR	0.6	90
22	NR	IM	FR	0.5	40
23	NR	IM	IM	0.2	25
24	NR	IM	NR	0.1	90
25	NR	NR	FR	0.4	30
26	NR	NR	IM	0.1	15
27	NR	NR	NR	0	90

6.5 实际性能评估

利用本书所提系统在真实室内环境下进行了几个实验，在此主要介绍四种示例情况。

1. 示例Ⅰ

在本示例中，要求机器人沿一条曲线运动。路径宽度小于机器人本体的宽度。图 6.10 给出了机器人执行该任务时的序列图像。图 6.10a ~ f 分别给出了机器人试图跟踪路径中线的图像序列。对于所有示例情况，采用视觉和红外测距的导航，因此在客户端关闭接近觉传感器。图 6.11 给出了细实线表示的完整运动路径。机器人的理想路径用路径中线上的点线表示。由图可知，机器人的实际运动路径与理想路径非常接近。只是在角点处，实际路径与理想路径的偏差稍大。这不难理解，在采用基于模糊的导航算法时，实际机器人应经过一个平滑的转向角。

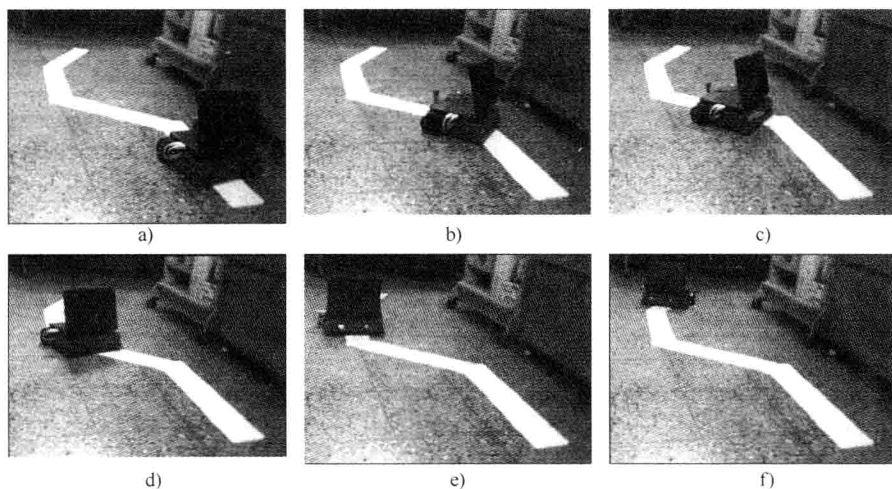


图 6.10　示例Ⅰ中机器人经过的路径

2. 示例Ⅱ

在本示例中，要求机器人沿一条折线运动，当路径结束时，大概是个 U 形，并沿路径返回到其初始起始点。图 6.12 给出了机器人执行该任务的图像序列。该示例研究证明当前向路径结束时，从远程客户端对机器人发出利用所保存的可能转向角绕行信息指令来自动转弯，并沿原路径返回。因此，在到达路径终点时，机器人无需停止，PixelCountLeft 和 PixelCountRight 计算得到零值。实际上，机器人以一种迭代方式不断转弯，直到其能重新跟踪路径。图 6.12a ~ f 分别给出了机器人前向运动时试图跟踪路径中线的序列图像。

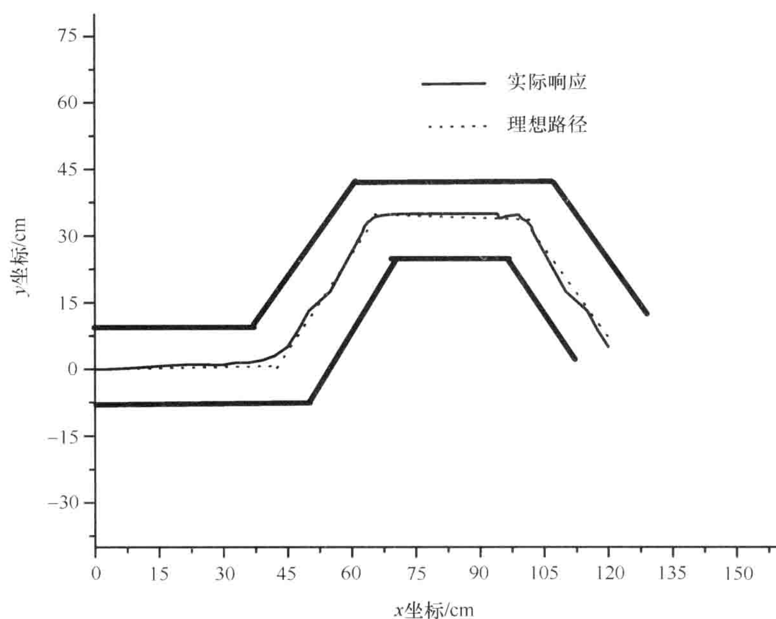


图 6.11 示例 I 中机器人经过的完整路径

图 6.12g ~ k 给出了机器人以迭代方式执行转弯操作，使其能重新定位路径中线时的序列图像。图 6.12l ~ o 给出了机器人可重新跟踪路径并返回其原始起

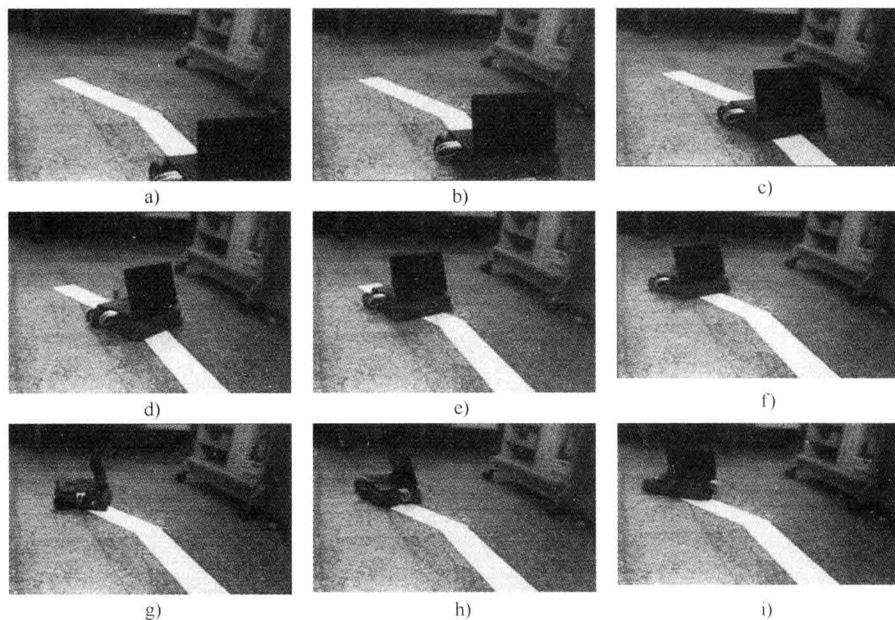
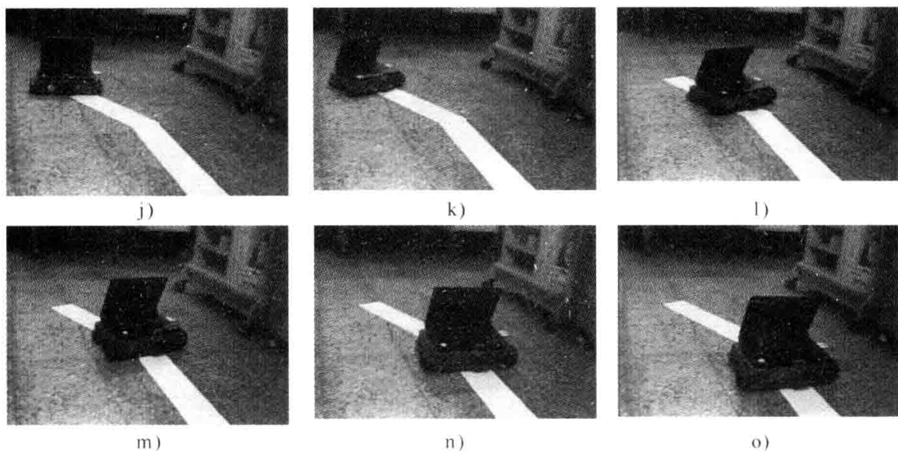


图 6.12 示例Ⅱ中机器人经过的路径



j)　k)　l)　m)　n)　o)

图 6.12　示例Ⅱ中机器人经过的路径(续)

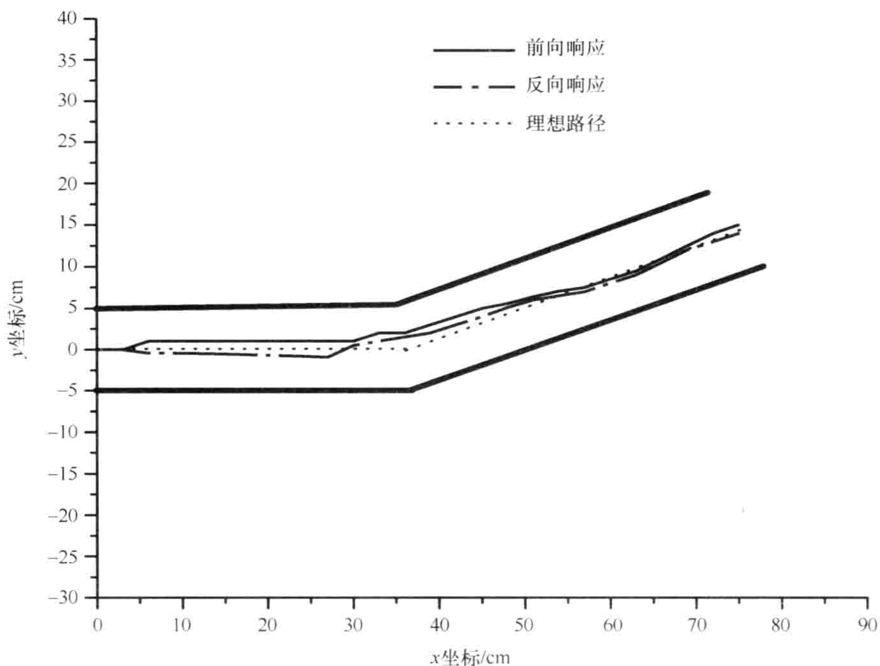


图 6.13　示例Ⅱ中机器人经过的完整路径

始点的序列图像。从客户端选择“Search ON”操作可使机器人试图重新跟踪路径，即便在视觉感知器视场中路径已消失时。图 6.13 给出了运动的前向路径(实线)。机器人的理想路径为沿路径中线的点线。可看出机器人实际运行的路径与理想路径非常接近，在角点处，实际路径与理想路径有一定偏差。另外，还可看出经过 U 形弯之后，用点画线表示的机器人运动路径与红线表示的前向

运动时的路径有较小偏差。这表明对于机器人前向和返回运动时都具有良好性能。

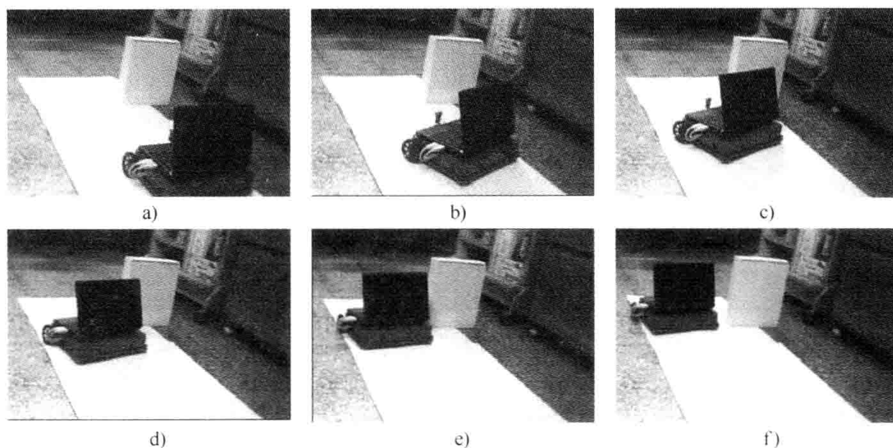


图 6.14 示例Ⅲ中机器人经过的路径

3. 示例Ⅲ

在本示例中，要求机器人尽可能地沿比机器人较宽的路径中线运动。在路径上有一个障碍物，机器人需要避开。图 6.14 给出了机器人如何执行任务的序列图像。在本实验中，要求机器人在从客户端以“Search OFF”操作下进行导航。

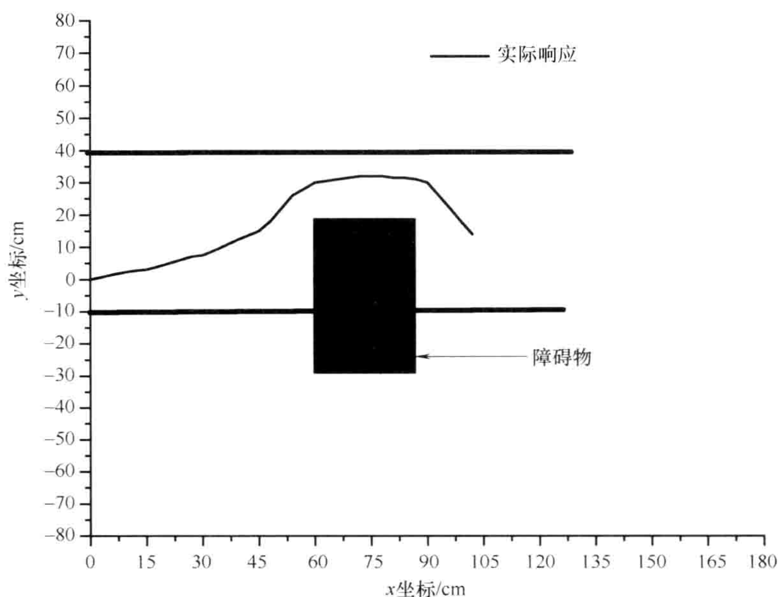


图 6.15 示例Ⅲ中机器人经过的完整路径

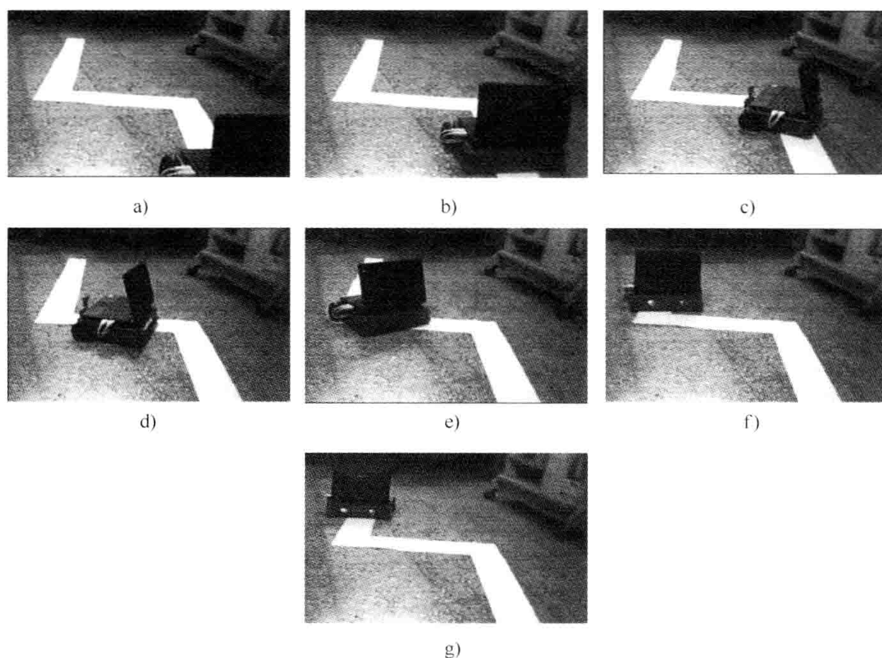


图6.16 示例Ⅳ中机器人经过的路径

因此，机器人在成功避障后到达路径终点时停止。正如要求所述，当摄像头视场中路径消失时，机器人无需在路径中重新定位自己。图6.15给出了细实线所示的完整运动路径。这表明机器人如何在开始时利用视觉感知沿路径中线运动，然后当遇到障碍物时，利用红外避障算法左转，与障碍物几乎平行地成功避障，然后当绕过障碍后，试图利用视觉感知来重新定位自身在路径中线。

4. 示例Ⅳ

在本示例中，要求机器人执行更困难的任务，机器人必须沿路线两个几乎垂直的转弯，并要求沿路径中线。在该示例中，要求机器人在从客户端以"Search OFF"操作下进行导航。图6.16给出了当机器人执行导航任务时的图像序列。由图可看出尽管是直角转弯，机器人仍能够在转弯后重新定位自身在路径中线，并以满意的形式沿路径运动。从图6.17中可看出，甚至经过两个直角转弯机器人还能够快速重新定位在路径中线，实际路径与理想路径之间的偏差相当小。该偏差在两个直角拐角处稍大，这再次证明了之前所述的逻辑。

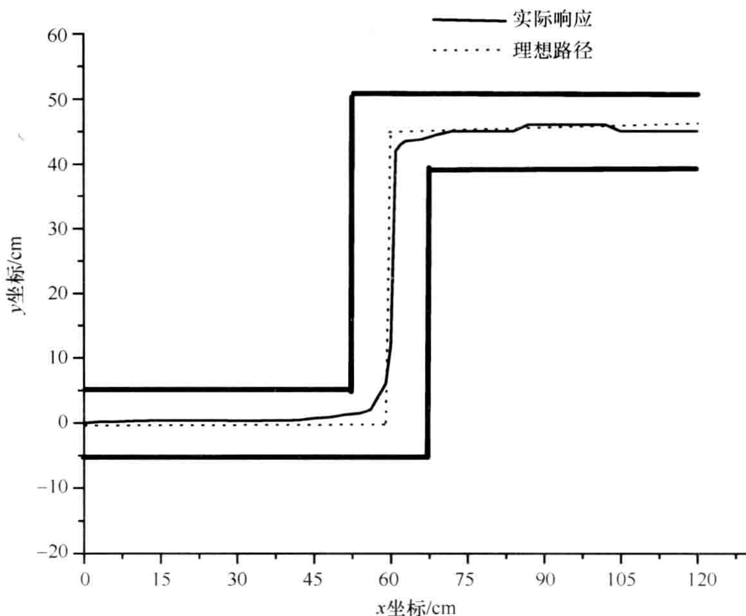


图 6.17 示例Ⅳ中机器人经过的完整路径

6.6 本章小结

本章讨论了在室内环境下如何实现一种简单而有效的基于模糊逻辑的路径跟踪机制，在本实验室自主开发的移动机器人上进行了实验。该机制是在机器人前方无障碍时，采用基于视觉的模糊导航；当前方有障碍时，采用基于红外的模糊避障，同时停用视觉导航机制。一旦完成避障，则停用红外避障，继续采用视觉导航。不管目标是沿着一条细线还是一条较宽的路径运动，该机制总是试图引导机器人沿着路径的中线运动。本系统能够实现在有障碍和无障碍下的导航，并成功尝试了内网连接功能。该移动机器人路径跟踪系统，结合本实验室自主开发的移动机器人平台上，在几种真实的室内环境下进行了实验验证。在此，讨论了四种实际示例来表明该系统的有用性和有效性。

致谢 本章所介绍的工作得到了印度大学教育资助委员会重大研究项目计划的资助[资助号：32-118/2006(SR)]。

参考文献

[1] Takeuchi, T., Nagai, Y., Enomoto, N.: Fuzzy control of a mobile robot for obstacle avoidance. Information Sciences 43, 231–248 (1988)

[2] Blochl, B.: Fuzzy Control in Real-Time for Vision Guided Autonomous Mobile Robots. In: Klement, E.-P., Slany, W. (eds.) FLAI 1993. LNCS, vol. 695, pp. 114–125. Springer, Heidelberg (1993)

[3] Yen, J., Pfluger, N.: A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation. IEEE Transactions on Systems, Man, and Cybernatics 25(6), 971–978 (1995)

[4] Pan, J., Pack, D.J., Kosaka, A., Kak, A.C.: FUZZY-NAV: A vision-based robot navigation architecture using fuzzy inference for uncertainty. In: Proc. IEEE World Congress Neural Networks, vol. 2, pp. 602–607 (July 1995)

[5] Maeda, M., Shimakawa, M., Murakami, S.: Predictive fuzzy control of an autonomous mobile robot with forecast learning function. Fuzzy Sets and Systems 72, 51–60 (1995)

[6] Congdon, C., Huber, M., Kortenkamp, D., Konolige, K., Myers, K., Ruspini, E.H., Saffiotti, A.: CARMEL vs. Flakey: A comparison of two winners. Artificial Intelligence Magazine 14(1), 49–57 (1993)

[7] Goodridge, S.G., Luo, R.C.: Fuzzy behavior fusion for reactive control of an autonomous mobile robot: MARGE. In: Proc. IEEE International Conference on Robotics and Automation, San Diego, pp. 1622–1627 (1994)

[8] Saffiotti, A.: The uses of fuzzy logic in autonomous robot navigation. Soft Computing 1, 180–197 (1997)

[9] Howard, A., Seraji, H., Tunstel, E.: A rule-based fuzzy traversability index for mobile robot navigation. In: Proc. International Conference on Robotics and Automation, Korea, pp. 3067–3071 (May 2001)

[10] Nirmal Singh, N.: Vision Based Autonomous Navigation of Mobile Robots. Ph.D. Thesis, Jadavpur University, Kolkata, India (2010)

第7章　移动机器人的SLAM㊀

摘要：本章首先介绍了移动机器人导航中的同时定位与地图构建（SLAM）概念，然后详细介绍了基于扩展卡尔曼滤波（EKF）的SLAM算法。接下来，在一个较为复杂的场景中，且传感器数据不正确的情况下来实现基于EKF的SLAM算法，并讨论了如何利用模糊或神经—模糊监督方法来提高这种情况下的估计性能。在此背景下，还讨论了如何利用进化优化策略来自动学习该神经—模糊监督器的自由参数。

7.1　简介

过去的20年中，SLAM问题在自主车辆和移动机器人等研究领域已得到广泛关注。从本质上来说，SLAM问题是指在一个未知环境中迭代估计一个移动平台的未知运动，由此来确定包含特征（也称为路标）的环境地图，并在各自信息的基础上，确定移动平台的绝对位置[1]。这是一个非常复杂的问题，因为车辆的位姿估计（pose estimate）及其相关的地图估计会由于缺少任何全局位置信息而导致越来越不精确[2]。若车辆没有配置全球定位系统（GPS），就会产生该问题。因此，SLAM问题的复杂性是多方面的，由于车辆位姿和地图估计之间的相互依赖关系，这需要在高维空间中进行求解[3]。

解决SLAM问题的一种最早也是最常用的方法是采用EKF技术。直到现在，还有大量研究工作是采用EKF方法来解决SLAM的相关问题[1,4-12]。针对室内[13,14]、室外[7]、水下[15]、地下[16]等环境，已成功开发了一些SLAM算法。一种基于EKF的方法是在环境地图中以一种完整状态矢量的形式来估计和保存机器人位姿和特征位置，并以误差协方差矩阵的形式保存这些估计的不确定性。这些协方差矩阵中还包括了表示特征/路标估计之间互相关性的元素。然而，在

㊀ 本章是基于：

1）Amitava Chatterjee 和 Fumitoshi Matsuno 于2007年10月在 IEEE Transactions on Fuzzy systems 期刊上第15卷第5期984～997页上发表的“A neuro-fuzzy assisted extended Kalman filter-based approach for Simultaneous Localization and Mapping problems”论文；

2）Amitava Chatterjee 于2009年5月在 Robotica 期刊上第27卷第3期411～423页上发表的“Differential evolution tuned fuzzy supervisor adapted extended kalman filtering for SLAM problems in mobile robots”。经剑桥大学出版社许可使用。

采用经典的完全基于 EKF 的 SLAM 方法时，一个主要问题是在具有大量特征的情况下，计算量会显著增大，这是由于总的状态矢量和总的协方差矩阵会急剧增大。之后，在对基于 EKF 的 SLAM 算法的改进研究中，将该问题看作一个主要的研究内容，目前已提出一些改进算法[7,9,17-19]。与基于 EKF 的 SLAM 算法相关的另一个关键问题是由于地图中一些形状相似的路标所造成的数据关联问题。在这种情况下，不同的数据关联假设都会产生多种完全不同的地图，并且无法利用高斯分布来表示这种多模态的分布。该问题通常是在给定当前机器人地图的条件下，在单次测量[1]或多次测量[20]的基础上通过限制算法只与最相似的数据进行关联来解决。其中，采用多次测量的方法更具有鲁棒性。尽管目前已开发出其他一些如参考文献[21，22]中的数据关联算法，但由于这些算法无法实时实现，因此意义不大。

另外还提出了一些解决 SLAM 问题的其他方法，这些方法都是通过某些数值算法来实现的，而不是采用类似于 EKF 中的严格统计方法。其中，某些方法基于贝叶斯理论，这样就无需 EKF 中的假设条件（即不确定性需由高斯分布进行建模）。采用 Sequential Monte Carlo（SMC）方法开发出几种这样的算法，其中需要进行粒子滤波[2,3,23,24]。通过在 EKF 算法中对非线性模型线性化来引入一个不确定性，即可不用粒子滤波技术。然而，在基于粒子滤波的方法中，希望利用大量粒子以使得能够包含一个在每个采样时刻与车辆/机器人所处位置非常接近的粒子[25]。如何开发一种有效的 SLAM 算法，并在粒子滤波中采用足够少的粒子数，仍是目前一个重要的研究内容。在该研究方向上，FastSLAM1.0 和 FASTSLAM2.0 算法是一个重要的飞跃，已成功解决基于粒子滤波的 SLAM 中的维数问题[26]。另外，还有一些 SLAM 算法采用扫描匹配技术得到成功地开发应用，即通过参考帧之间的一个空间关系图来有效地构建地图[7,27]。

之前已表明，EKF 的处理性能很大程度上取决于过程噪声协方差矩阵（$\boldsymbol{Q}$）和测量噪声协方差矩阵（$\boldsymbol{R}$）的准确性。$\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 的先验知识不正确可能会导致性能下降[28]，甚至可能会导致发散[29]。因此，自适应估计这些矩阵对于在线配置非常重要。在参考文献[28]中，Mehra 提出了一种对卡尔曼滤波算法中噪声协方差矩阵 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 自适应估计的开创性研究，该方法是基于能够对 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 进行渐进正态、无偏一致性估计的创新相关性分析方法[35]，具体算法是基于一种噪声概率统计固定不变且所考虑的模型是一个时不变模型的假设。后来，在这一研究方向上也进行了一些研究工作，采用的是一些自适应估计 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 的经典方法[30-35]。在参考文献[30]中，提出了一种迭代算法和随机逼近算法相结合的方法来估计 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$。在参考文献[32]和[33]中，将问题扩展到对 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 的时变估计。参考文献[34]对参考文献[28]中的实际应用进行了详细阐述。

在最近 10 年，还提出了一些基于模糊或模糊神经网络技术的自适应 EKF 算

法[36-39]。在参考文献[38]中，通过采用一个模糊神经网络确定 ARMA 模型中每个工作点的 AR 参数，然后对系统的等效状态空间表示进行卡尔曼滤波，来求解一个输入—输出的映射问题，其中输出变量会受测量噪声的影响。在参考文献[36]中，利用模糊逻辑来同时自适应估计 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$。在参考文献[37]中，对于卡尔曼滤波算法，利用模糊逻辑来仅估计 $\boldsymbol{R}$ 矩阵。在实际情况下，无法准确已知这些以 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 矩阵形式存在的信息矩阵是相当敏感的。这样就会在很大程度上影响 SLAM 问题的性能。

本章首先详细介绍基于 EKF 的随机 SLAM 算法。接下来，将探讨无法准确已知传感器噪声统计信息情况下的 SLAM 问题。在这些情况下，将讨论在模糊神经网络的作用下，基于 EKF 的算法是如何得到有效应用的[44,45]。在此，详细介绍了如何利用一个模糊神经网络模型来协助基于 EKF 的 SLAM 算法在每次迭代过程中自适应地估计 $\boldsymbol{R}$ 矩阵。另外，还将讨论如何利用常用的进化优化算法[如粒子群优化(PSO)算法[40]和差分进化算法]来学习模糊神经网络模型中的自由参数。本章所讨论的模糊自适应 EKF 算法基本上可实现一个更加复杂的系统，与之前的系统相比，主要表现在两个方面：

1）对于 SLAM 问题，由于状态矢量和协方差矩阵的大小本质上是时变的，因此这种情况基本上是非常复杂的。这是因为在导航过程中，不同时刻下在状态矢量中初始化新的路标(在某些特殊条件下，一些已存在的路标可能会被去除)，由此造成这些矢量和矩阵的大小会不断变化。通常情况下，这些矩阵的大小会逐渐增大。

2）本章所讨论的方法是采用了一种自学习模糊神经网络模型的通用方法。这与之前开发的手动选择模糊系统参数的系统形成鲜明对比。

本章对这些 SLAM 算法进行了详细深入的分析。在此，针对各种不同的环境条件(即特征/路标点的个数发生变化以及一些无法准确获知的测量噪声统计值)，给出了实验结果。

7.2 基于 EKF 的随机 SLAM 算法

1. 假设条件

1）假设所考虑的特征均为二维点特征；

2）假设这些特征均为静态特征，即在所建立的地图中，特征位置不会随时间变化；

3）在输入控制量、转向角控制指令(s)和后轮驱动速度(w)中存在不确定性，并且这些不确定性可由高斯分布进行建模；

4）假设机器人的起始位姿是确定的；

5）在两个连续采样时刻的间隔内，假设机器人的运动增量是线性的；

6）在距离(r)测量和朝向(θ)测量中具有不确定性，同样这些不确定性也由高斯分布进行建模；

7）这些特征属性只有其二维位置，在此不考虑其他属性，如形状等。

2. 算法

现在对采用 EKF 算法的基于特征地图的 SLAM 问题进行简要概述。对该算法的详细阐述可参见参考文献[6，7]。这些过程是在通过非线性动力学约束和/或涉及非线性测量关系的情况下，可采用 EKF 算法来进行状态估计的。在该方法中，对于根据测量所得的状态估计连续更新的滤波器估计轨迹进行线性化[43]。利用一个非线性函数 $\boldsymbol{f}(\cdot)$ 对状态转移进行建模，同时，状态的观测值或测量值也可由一个非线性函数 $\boldsymbol{h}(\cdot)$ 建模，如下所示：

$$\boldsymbol{x}_{k+1}=\boldsymbol{f}(\boldsymbol{x}_k,\ \boldsymbol{u}_k)+\boldsymbol{q}_k \tag{7.1}$$

$$\boldsymbol{z}_{k+1}=\boldsymbol{h}(\boldsymbol{x}_{k+1})+\boldsymbol{r}_{k+1} \tag{7.2}$$

式中　$\boldsymbol{x}_k$——采样时刻 k 时的($n\times1$)个过程状态矢量；

$\boldsymbol{z}_k$——采样时刻 k 时的($m\times$)个测量矢量；

$\boldsymbol{u}_k$——输入控制量；

$\boldsymbol{q}_k$ 和 $\boldsymbol{r}_k$——随机变量，分别表示高斯白噪声和测量噪声，$\boldsymbol{P}_k$、$\boldsymbol{Q}_k$ 和 $\boldsymbol{R}_k$ 分别表示 $\boldsymbol{x}_k$、$\boldsymbol{q}_k$ 和 $\boldsymbol{r}_k$ 的协方差矩阵。

在 SLAM 问题中，状态矢量 $\boldsymbol{x}$ 由车辆状态 $\boldsymbol{x}_{\mathrm{v}}$ 和路标状态 $\boldsymbol{x}_{\mathrm{m}}$ 组成。因此，对以均值矢量 $\hat{\boldsymbol{x}}$ 形式表示的总的状态矢量 $\boldsymbol{x}$ 以及相应的总的误差协方差矩阵 $\boldsymbol{P}$ 的估计为

$$\hat{\boldsymbol{x}}=[\hat{\boldsymbol{x}}_{\mathrm{v}}^{\mathrm{T}},\ \hat{\boldsymbol{x}}_{\mathrm{m}}^{\mathrm{T}}]^{\mathrm{T}} \tag{7.3}$$

$$\boldsymbol{P}=\begin{bmatrix}\boldsymbol{P}_{\mathrm{v}} & \boldsymbol{P}_{\mathrm{vm}}\\ \boldsymbol{P}_{\mathrm{vm}}^{\mathrm{T}} & \boldsymbol{P}_{\mathrm{m}}\end{bmatrix} \tag{7.4}$$

式中　$\hat{\boldsymbol{x}}_{\mathrm{v}}$——机器人/车辆状态的均值估计（由其位姿表示）；

$\boldsymbol{P}_{\mathrm{v}}$——与 $\hat{\boldsymbol{x}}_{\mathrm{v}}$ 相关联的误差协方差矩阵；

$\hat{\boldsymbol{x}}_{\mathrm{m}}$——特征位置的均值估计；

$\boldsymbol{P}_{\mathrm{m}}$——与 $\hat{\boldsymbol{x}}_{\mathrm{m}}$ 相关联的误差协方差矩阵。

在一个任意直角坐标系中定义机器人/车辆位姿。认为特征或路标是一个二维的点特征。假设在地图中观测到 n 个这样的静态点特征，则

$$\hat{\boldsymbol{x}}_{\mathrm{v}}=[\hat{\boldsymbol{x}}_{\mathrm{v}},\ \hat{\boldsymbol{y}}_{\mathrm{v}},\ \hat{\boldsymbol{\varphi}}_{\mathrm{v}}]^{\mathrm{T}} \tag{7.5}$$

$$
\boldsymbol{P}_{\mathrm{v}}=\begin{bmatrix}\sigma_{x_vx_v}^2 & \sigma_{x_vy_v}^2 & \sigma_{x_v\varphi_v}^2\\ \sigma_{x_vy_v}^2 & \sigma_{y_vy_v}^2 & \sigma_{y_v\varphi_v}^2\\ \sigma_{x_v\varphi_v}^2 & \sigma_{y_v\varphi_v}^2 & \sigma_{\varphi_v\varphi_v}^2\end{bmatrix} \tag{7.6}
$$

$$
\hat{\boldsymbol{x}}_{\mathrm{m}}=[\hat{\boldsymbol{x}}_1\ \hat{\boldsymbol{y}}_1\cdots\hat{\boldsymbol{x}}_n\ \hat{\boldsymbol{y}}_n]^{\mathrm{T}} \tag{7.7}
$$

$$
\boldsymbol{P}_{\mathrm{m}}=\begin{bmatrix}\sigma_{x_1x_1}^2 & \sigma_{x_1y_1}^2 & \cdots & \sigma_{x_1x_n}^2 & \sigma_{x_1y_n}^2\\ \sigma_{x_1y_1}^2 & \sigma_{y_1y_1}^2 & \cdots & \sigma_{y_1x_n}^2 & \sigma_{y_1y_n}^2\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \sigma_{x_1x_n}^2 & \sigma_{y_1x_n}^2 & \cdots & \sigma_{x_nx_n}^2 & \sigma_{x_ny_n}^2\\ \sigma_{x_1y_n}^2 & \sigma_{y_1y_n}^2 & \cdots & \sigma_{x_ny_n}^2 & \sigma_{y_ny_n}^2\end{bmatrix} \tag{7.8}
$$

根据这些静态特征的位置估计来定义地图，并且式(7.4)的 $\boldsymbol{P}_{\mathrm{vm}}$ 中保持了机器人与地图之间的关系。$\boldsymbol{P}_{\mathrm{m}}$ 中的非对角线元素体现了互相关性，因此特征本身之间的信息是无关的。假设系统初始时没有观测特征，直角坐标系与机器人的起始位姿一致，并且机器人的起始位姿不存在不确定性。用数学公式可表示为 $\hat{\boldsymbol{x}}=\hat{\boldsymbol{x}}_{\mathrm{v}}=\boldsymbol{0}$ 和 $\boldsymbol{P}=\boldsymbol{P}_{\mathrm{v}}=\boldsymbol{0}$。

随着机器人开始运动，$\hat{\boldsymbol{x}}_{\mathrm{v}}$ 和 $\boldsymbol{P}_{\mathrm{v}}$ 也变为非零值。在随后的迭代中，进行第一次观测时，要求对新特征初始化，并首次出现 $\hat{\boldsymbol{x}}_{\mathrm{m}}$ 和 $\boldsymbol{P}_{\mathrm{m}}$。这就会增大 $\hat{\boldsymbol{x}}$ 和 $\boldsymbol{P}$ 的大小，并重新计算矢量 $\hat{\boldsymbol{x}}$ 和矩阵 $\boldsymbol{P}$ 的元素。不断迭代执行该过程。

3. 时间更新(“预测”)步骤

在此，假设在机器人运动的影响下，输入控制矢量 $\boldsymbol{u}$ 包括两个控制输入，即转向角指令($\boldsymbol{s}$)和后轮驱动速度($\boldsymbol{w}$)。因此，$\boldsymbol{u}=[w\ s]^{\mathrm{T}}$。这样，状态估计就可通过编码器里程计和机器人运动模型获得。在此，必须要考虑输入控制量 w 和 s 的不确定性(如由于车轮打滑、控制器校正不准等引起的不确定性)，并以 w 和 s 的额定值来进行一种改进的高斯分布建模。因此，在预测步骤中从采样时刻 k 到$(k+1)$的间隔内计算状态估计的预测值和误差协方差的估计值，即

$$
\hat{\boldsymbol{x}}_{k+1}^{-}=\boldsymbol{f}(\hat{\boldsymbol{x}}_k,\ \hat{\boldsymbol{u}}_k)=\begin{bmatrix}\hat{\boldsymbol{x}}_{\mathrm{v}_{k+1}}^{-}\\ \hat{\boldsymbol{x}}_{\mathrm{m}}\end{bmatrix}=\begin{bmatrix}\hat{\boldsymbol{f}}_{\mathrm{v}}(\hat{\boldsymbol{x}}_{\mathrm{v}_k},\ \hat{\boldsymbol{u}}_k)\\ \hat{\boldsymbol{x}}_{\mathrm{m}}\end{bmatrix} \tag{7.9}
$$

$$\boldsymbol{P}_{k+1}^{-}=\begin{bmatrix}\nabla \boldsymbol{f}_{\boldsymbol{x}_{vk}}\boldsymbol{P}_{v_k}\nabla \boldsymbol{f}_{\boldsymbol{x}_{vk}}^{T}+\nabla \boldsymbol{f}_{\boldsymbol{u}_k}\boldsymbol{U}_k\nabla \boldsymbol{f}_{\boldsymbol{u}_k}^{T} & \nabla \boldsymbol{f}_{\boldsymbol{x}_{vk}}\boldsymbol{P}_{vm_k}\\ (\nabla \boldsymbol{f}_{\boldsymbol{x}_{vk}}\boldsymbol{P}_{vm_k})^{T} & \boldsymbol{P}_{m}\end{bmatrix} \tag{7.10}$$

其中，在运动模型和输入控制量的基础上，用 $\boldsymbol{f}_v$ 来估计机器人位姿。根据在此采用的移动机器人的里程计方程，即假设机器人的增量运动本质上是线性的，这样 $\boldsymbol{f}_v$ 可表示为[42]

$$\hat{\boldsymbol{x}}_{v_{k+1}}^{-}=\begin{bmatrix}\hat{x}_{v_{k+1}}^{-}\\ \hat{y}_{v_{k+1}}^{-}\\ \hat{\varphi}_{v_{k+1}}^{-}\end{bmatrix}=\boldsymbol{f}_v(\hat{\boldsymbol{x}}_{v_k},\ \hat{\boldsymbol{u}}_k)=\begin{bmatrix}\hat{x}_{v_k}+w_k\times\Delta t\times\cos(s_k+\hat{\varphi}_{v_k})\\ \hat{y}_{v_k}+w_k\times\Delta t\times\sin(s_k+\hat{\varphi}_{v_k})\\ \hat{\varphi}_{v_k}+w_k\times\Delta t\times\dfrac{\sin(s_k)}{WB}\end{bmatrix} \tag{7.11}$$

式中　WB——机器人的轴距；

　　Δt——采样时刻。

相应的雅各比矩阵、$\boldsymbol{U}_k$ 和 $\boldsymbol{u}$ 的协方差矩阵可记为

$$\nabla \boldsymbol{f}_{\boldsymbol{x}_{vk}}=\frac{\partial \boldsymbol{f}_{v_k}}{\partial \boldsymbol{x}_{v_k}}\Big|(\hat{\boldsymbol{x}}_{v_k},\hat{\boldsymbol{u}}_k) \tag{7.12}$$

$$\nabla \boldsymbol{f}_{\boldsymbol{u}_k}=\frac{\partial \boldsymbol{f}_{v_k}}{\partial \boldsymbol{u}_k}\Big|(\hat{\boldsymbol{x}}_{v_k},\hat{\boldsymbol{u}}_k) \tag{7.13}$$

$$\boldsymbol{U}=\begin{bmatrix}\sigma_v^2 & 0\\ 0 & \sigma_s^2\end{bmatrix} \tag{7.14}$$

在此，由于假设这些特征均为静态特征，因此式(7.9)和式(7.10)中的 $\hat{\boldsymbol{x}}_m$ 和 $\boldsymbol{P}_m$ 为时不变常数。

4. 测量值更新(“校正”)步骤

在此，假设已观测到特征地图中的一个特征，其位置表示为第 i 个特征，即 $(\hat{\boldsymbol{x}}_i,\ \hat{\boldsymbol{y}}_i)$。对于所考虑的系统[7,42]，假设是利用一个二维激光测距扫描仪(SICK PLS)来观测特征，该激光测距仪是目前在移动机器人导航中最常用的测距传感器。假设安装在移动机器人前部防碰装置上的该激光测距扫描仪能够以 0.5°的间隔返回在 180°平面上扫描得到的距离值，这种测距传感器的精度通常大约为 ±50mm。值得注意的是，还需假设移动机器人上已配置有车轮和转向编码器。表示每个特征与激光扫描仪(即移动机器人)之间相对距离的距离测量值是以极坐标形式显示的。设特征是由其相对于观测者的距离(r)和角度(θ)来表征的，

记为

$$z = [r\theta]^{\mathrm{T}} \tag{7.15}$$

这些观测值中的不确定性也是由一种改进的高斯分布来建模的。设 $\boldsymbol{R}$ 为相应的观测/测量噪声协方差矩阵，记为

$$\boldsymbol{R} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \tag{7.16}$$

在此，假设距离测量值和角度测量值之间没有互相关性。在地图中，这些测量值可记为

$$\hat{\boldsymbol{z}}_{i_k} = \boldsymbol{h}_i(\hat{\boldsymbol{x}}_k) = \begin{bmatrix} \sqrt{(\hat{x}_i - \hat{x}_{\mathrm{v}_k})^2 + (\hat{y}_i - \hat{y}_{\mathrm{v}_k})^2} \\ \arctan\left(\dfrac{\hat{y}_i - \hat{y}_{\mathrm{v}_k}}{\hat{x}_i - \hat{x}_{\mathrm{v}_k}}\right) - \hat{\varphi}_{\mathrm{v}_k} \end{bmatrix} \tag{7.17}$$

现在，假设 z 和$(\hat{x}_i, \hat{y}_i)$正确关联，由此可计算 EKF 的增益 $\boldsymbol{W}_i$，相应的计算公式为

$$\boldsymbol{v}_{i_{k+1}} = \boldsymbol{z}_{k+1} - \boldsymbol{h}_i(\hat{\boldsymbol{x}}_{k+1}^-) \tag{7.18}$$

$$\boldsymbol{S}_{i_{k+1}} = \nabla \boldsymbol{h}_{\boldsymbol{x}_{k+1}} \boldsymbol{P}_{k+1}^- \nabla \boldsymbol{h}_{\boldsymbol{x}_{k+1}}^{\mathrm{T}} + \boldsymbol{R}_k \tag{7.19}$$

$$\boldsymbol{W}_{i_{k+1}} = \boldsymbol{P}_{k+1}^- \nabla \boldsymbol{h}_{\boldsymbol{x}_{k+1}}^{\mathrm{T}} \boldsymbol{S}_{i_{k+1}}^{-1} \tag{7.20}$$

式中　$\boldsymbol{v}_i$——第 i 个路标的观测新息(in novation)；

$\boldsymbol{S}_i$——关联新息的协方差矩阵。

雅各比矩阵$\nabla \boldsymbol{h}_{\boldsymbol{x}_{k+1}}$可表示为

$$\nabla \boldsymbol{h}_{\boldsymbol{x}_{k+1}} = \frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{x}_k} \Big|_{\hat{\boldsymbol{x}}_{k+1}^-} \tag{7.21}$$

因此，后验增广状态估计和相应的协方差矩阵可按如下公式进行更新：

$$\hat{\boldsymbol{x}}_{k+1}^+ = \hat{\boldsymbol{x}}_{k+1}^- + \boldsymbol{W}_{i_{k+1}} \boldsymbol{v}_{i_{k+1}} \tag{7.22}$$

$$\boldsymbol{P}_{k+1}^+ = \boldsymbol{P}_{k+1}^- - \boldsymbol{W}_{i_{k+1}} \boldsymbol{S}_{i_{k+1}} \boldsymbol{W}_{i_{k+1}}^{\mathrm{T}} \tag{7.23}$$

在此，需要注意的是，除了过程和测量中的不确定性，EKF 算法中线性化处理也会产生不确定性。通过对非线性函数 $\boldsymbol{f}(\cdot)$和 $\boldsymbol{h}(\cdot)$在状态平均点处进行线性化可得到“时间更新”方程和“测量更新”方程。具体的线性化处理是通过泰勒级数展开并忽略高阶项来进行的。这种通过一阶偏导对非线性系统进行近似化处理的方式会在 EKF 算法中引入额外的不确定性。实际上，对于高阶非线性函数，这种线性化处理方式并不能充分准确地近似协方差转换，这也许会导致非常不一致的不确定性估计。在这种情况下，无迹变换或许能提供更准确的结

果。

5. 初始化新特征并删除旧特征

在执行递归预测和更新的迭代过程中，随时都可能观测到新特征。然后，应通过将这些新特征位置的二维坐标集成到增广状态矢量中来将其初始化到系统，并相应地改变协方差矩阵。由激光扫描仪观测到的特征可能是点、线、角点和边等。在此，仅考虑这些特征为点特征时的情况，每个点特征都表示二维环境地图中的一个唯一点。按照参考文献[7]中的数学计算公式，这些新的$\hat{\boldsymbol{x}}_k^+$和$\boldsymbol{P}_k^+$可计算为

$$\hat{\boldsymbol{x}}_k^+ = \begin{bmatrix} \hat{\boldsymbol{x}}_k \\ \boldsymbol{f}_f(\hat{\boldsymbol{x}}_{v_k},\ \boldsymbol{z}_k) \end{bmatrix} \tag{7.24}$$

$$\boldsymbol{P}_k^+ = \begin{bmatrix} \boldsymbol{P}_{v_k} & \boldsymbol{P}_{vm_k} & \boldsymbol{P}_{v_k}\nabla\boldsymbol{f}_{f_{vk}}^{\mathrm{T}} \\ \boldsymbol{P}_{vm_k}^{\mathrm{T}} & \boldsymbol{P}_{m} & \boldsymbol{P}_{vm_k}^{\mathrm{T}}\nabla\boldsymbol{f}_{f_{vk}}^{\mathrm{T}} \\ \nabla\boldsymbol{f}_{f_{vk}}\boldsymbol{P}_{v_k} & \nabla\boldsymbol{f}_{f_{vk}}\boldsymbol{P}_{vm_k} & \nabla\boldsymbol{f}_{f_{vk}}\boldsymbol{P}_{v_k}\nabla\boldsymbol{f}_{f_{vk}}^{\mathrm{T}} + \nabla\boldsymbol{f}_{f_{zk}}\boldsymbol{R}_k\nabla\boldsymbol{f}_{f_{zk}}^{\mathrm{T}} \end{bmatrix} \tag{7.25}$$

在此，$\boldsymbol{f}_f(\hat{\boldsymbol{x}}_v,\ \boldsymbol{z})$是将极坐标观测值$\boldsymbol{z}$转换到直角坐标系下。雅各比矩阵可计算为

$$\nabla\boldsymbol{f}_{f_{vk}} = \frac{\partial \boldsymbol{f}_f}{\partial \boldsymbol{x}_{v_k}} \mid (\hat{\boldsymbol{x}}_{v_k}, \boldsymbol{z}_k)$$

$$\nabla\boldsymbol{f}_{f_{zk}} = \frac{\partial \boldsymbol{f}_f}{\partial \boldsymbol{z}_k} \mid (\hat{\boldsymbol{x}}_{v_k}, \boldsymbol{z}_k) \tag{7.26}$$

删除不可靠的特征相对简单。仅需从状态矢量中删除相关的行矢量，并从协方差矩阵中删除相关的行矢量和列矢量即可。

现在，在执行观测步骤时，同一时刻多个路标可见是非常普遍的，因此需要进行几次独立的观测。在本系统中，假设执行一次观测就可得到一批这样的观测值(即$\boldsymbol{z} = [r_1,\ \theta_1,\ \cdots,\ r_n,\ \theta_n]^{\mathrm{T}}$)，并进行批量更新。参考文献[7]表明对于 SLAM 算法，如果新息矢量$\boldsymbol{v}$中同时包含多个观测，则 EKF 算法具有更好的更新性能。在此，与参考文献[7]的论述一致。因此，在批量模式的观测和更新过程中，相应的 SLAM 算法是基于$\boldsymbol{v}$、$\boldsymbol{S}$和$\boldsymbol{W}$矢量/矩阵的组合，并且这三个矢量/矩阵的大小是随时间变化的，这是因为在任意观测时刻，可见路标的总个数是不断变化的。

7.3 结合模糊神经网络模型的基于 EKF 的 SLAM 算法

目前，自适应 EKF 领域中的大多数研究工作都主要关注于利用新息序列中新的统计信息来校正状态估计。本书中所采用的自适应 EKF 方法是基于一种新息自适应估计(IAE)方法。IAE 方法最初在参考文献[28]中提出，随后在参考文献[37]中与模糊逻辑相结合来应用。该方法的基本思想是首先确定任意第 k 个时刻的新测量值 $\boldsymbol{z}_k$ 与相应预测估计值 $\hat{\boldsymbol{z}}_k$ 之差，然后利用该信息来校正已完成的估计/预测。自适应策略的目标是减少新息序列($\boldsymbol{S}_k$)的理想协方差与相应的实际协方差 $\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$ 之间的不一致。在本书的 SLAM 算法中，根据式(7.19)计算 $\boldsymbol{S}_k$，其中等式右边符合批处理观测和更新的思想。$\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$ 可计算为

$$\hat{\boldsymbol{C}}_{\mathrm{Inn}k}=\boldsymbol{v}_k\boldsymbol{v}_k^{\mathrm{T}} \tag{7.27}$$

式中 $\boldsymbol{v}_k$——符合批量模式的增广新息序列。

根据参考文献[37]，应根据在一个合适的运动估计窗口(大小为 M)中对 $\boldsymbol{v}_k\boldsymbol{v}_k^{\mathrm{T}}$ 运动平均来计算协方差。然而对于 SLAM 问题，增量 $\boldsymbol{v}_k$ 的大小随时间不断变化，这是由于应取决于任意一个观测和更新步骤中之前至少观测到一次的路标个数。因此，利用式(7.27)来计算 $\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$，而不是采用一种运动平均。由此，可计算在第 k 个时刻的不一致为

$$\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}=\hat{\boldsymbol{C}}_{\mathrm{Inn}k}-\boldsymbol{S}_k \tag{7.28}$$

在此，目标采用模糊逻辑来最小化该不一致，通过对 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$ 矩阵中每个对角元素都采用一个单输入单输出的模糊神经网络系统来实现。利用这些模糊规则来调整 $\boldsymbol{R}$ 矩阵，使得传感器统计适应于不一致矩阵 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$ 中的减小。一个结合模糊神经网络的基于 EKF 的完整 SLAM 算法如算法 7.1 所示。设系统中连续输入控制信号的采样时间为 25ms。

算法 7.1 结合模糊神经网络的基于 EKF 的 SLAM 算法

由算法 7.1 可知，每个 NFS 都采用一种非线性映射形式：$\Delta\boldsymbol{R}(j,j)=f_{\mathrm{NFS}}(\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}(j,j))$，其中 $\Delta\boldsymbol{R}(j,j)$ 对应于增广测量噪声协方差矩阵 $\boldsymbol{R}$ 中相应对角元素的推荐值，并根据批量模式计算。在每次迭代执行观测和更新步骤时计算该增广矩阵，其大小取决于观测步骤中所观测到的总的路标个数。为保证与之前至少观测过一次的批量路标一致，该增广矩阵 $\boldsymbol{R}$ 的大小为 $[2z_f\times 2z_f]$，其中 z_f 为在本次迭代中所观测到的路标数。利用初始 $[2\times 2]$ $\boldsymbol{R}$ 矩阵来构成该增广矩阵 $\boldsymbol{R}$，可表示为

$$
\boldsymbol{R}=\begin{bmatrix}
\sigma_r^2 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \sigma_\theta^2 & \cdots & \cdots & \cdots & 0 & 0\\
\vdots & \vdots & \sigma_r^2 & 0 & \cdots & \vdots & \vdots\\
\vdots & \vdots & 0 & \sigma_\theta^2 & \cdots & \vdots & \vdots\\
\vdots & \vdots & \cdots & \cdots & \ddots & \vdots & \vdots\\
0 & 0 & \cdots & \cdots & \cdots & \sigma_r^2 & 0\\
0 & 0 & \cdots & \cdots & \cdots & 0 & \sigma_\theta^2
\end{bmatrix} \tag{7.29}
$$

其中，σ_r^2 和 σ_θ^2 对应于本次迭代所计算的传感器统计值。由此可看出，增广矩阵 $\boldsymbol{R}$ 仅由对角元素组成，而所有非对角元素都为零。这与 7.2 节中的假设相一致，即距离测量值和方位测量值相互独立，之间没有互相关性。在不同的迭代中，增广矩阵 $\boldsymbol{R}$ 的大小不断变化，这是因为，在某次迭代过程中，之前已访问过现又被观测到的路标个数会随着迭代的执行而变化。增广矩阵 $\boldsymbol{R}$ 的维数与 $\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$ 以及 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$ 一致。

根据对增广矩阵 $\boldsymbol{R}$ 中每个对角元素都采用相同的 NFS 来实现的思想，对每个 NFS 都采用归一化的输入。在实际中，NFS 采用下列三种形式的模糊 IF-THEN 规则：

IF($\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}(j,\ j)$)为负　　THEN$\Delta\boldsymbol{R}(j,\ j)=w_1$，

IF($\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}(j,\ j)$)为零　　THEN$\Delta\boldsymbol{R}(j,\ j)=w_2$，

IF($\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}(j,\ j)$)为正　　THEN$\Delta\boldsymbol{R}(j,\ j)=w_3$。

w_1、w_2 和 w_3 表示 $\Delta\boldsymbol{R}$ 矩阵中对角元素的模糊适应度推荐值，这取决于 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$ 矩阵中相应对角元素模糊化不一致的特性。然而对于距离观测和方位观测，不匹配的顺序可能会不同，这取决于分别已知的距离观测和方位观测传感器统计值的不准确性(或准确性)程度。因此，在按照算法 7.1 正确计算 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k_range_mismatch}$ 和 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k_bearing_mismatch}$ 的基础上，分别采用距离观测和方位观测的归一化输入。然后利用这些归一化输入，NFS 可计算每个对角元素的 $\Delta\boldsymbol{R}(j,\ j)$。最后计算适应度，即初始[$2\times2$]$\boldsymbol{R}$ 矩阵所需的 $\Delta\sigma_r^2$ 和 $\Delta\sigma_\theta^2$，在均值正确的基础上，分别计算 $\Delta\boldsymbol{R}(j,\ j)$数组中距离观测和方位观测的相关元素。当在执行 EKF 中的观测和更新步骤，并将用于随后形成的增广矩阵 $\boldsymbol{R}$ 等情况下，在下一次迭代中，调整后的初始矩阵[2×2]$\boldsymbol{R}$ 已准备好。然后，如果在观测步骤中观测到新特征时，则通过采用式(7.24)~式(7.26)来增广状态矢量和相应的协方差矩阵，从而执行每次观测和更新的步骤。

1. IF 已经过所有导航点，THEN 停止 ENDIF。

2. 计算机器人与当前导航点之间的距离。

IF(距离 < 所允许的最小距离)，

THEN 切换下一个导航点作为当前导航点 ENDIF。

3. 计算与当前导航点的转向角变化量(Δs)，然后得到新的转向角(s)[满足最大转向角变化率(Δs_{max})和最大转向角(s_{max})的约束条件]。

4. 机器人运动并确定其实际位姿。

5. 按照式(7.7)~式(7.10)，执行 EKF 预测步骤。

6. IF(Time_for_Observation 为真)，THEN 跳转到步骤 7，ELSE 跳转到步骤 1，ENDIF。

7. 确定从机器人当前的实际位置所观察到的可见路标集合。计算每个路标的实际距离值和方位角。根据已观测到的路标和新观测到的路标，分别进行观测。

8. 在第 5 步预测的总状态矢量增量的基础上，对步骤 7 中已观测到的路标预测距离—方位值。

9. 在实际和预测值的基础上，采用式(7.14)，对已观测到的路标批处理计算增广新息序列($\boldsymbol{\nu}$)。

10. 批处理计算相应的增广测量噪声协方差矩阵 $\boldsymbol{R}$(利用原始[2×2]$\boldsymbol{R}$ 矩阵)和增广线性观测模型 $\boldsymbol{h}$。

11. 在增广 $\boldsymbol{R}$ 和 $\boldsymbol{h}$ 的基础上，利用式(7.15)批处理计算增广 $\boldsymbol{S}$。

12. 根据式(7.18)和式(7.19)，更新后验状态估计矢量和误差协方差矩阵。

13. 根据式(7.23)和式(7.24)，分别计算$\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$和 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$，并确定 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$的大小，即[$\Delta\hat{\boldsymbol{C}}_{\mathrm{rows}}$，$\Delta\hat{\boldsymbol{C}}_{\mathrm{cols}}$]。

14. 分别确定 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}$矩阵相应对角线元素的距离观测误匹配的最大绝对值($\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k_\mathrm{range}_\mathrm{mismatch}}$)和方位观测误匹配的最大绝对值($\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k_\mathrm{bearing}_\mathrm{mismatch}}$)。

15. FOR 对于 $\Delta\hat{\boldsymbol{C}}_{\mathrm{rows}} j=1$，

用合适的 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k_\mathrm{range}_\mathrm{mismatch}}$或 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k_\mathrm{bearing}_\mathrm{mismatch}}$归一化相应的对角线元素 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}(j, j)$。

确定 NFS 输出的相应 $\Delta\boldsymbol{R}(j, j)$以及输入的归一化 $\Delta\hat{\boldsymbol{C}}_{\mathrm{Inn}k}(j, j)$。

ENDFOR

16. 将 $\Delta\sigma_r^2$ 作为对应于距离测量的 $\Delta\boldsymbol{R}(j, j)$元素的均值。

17. 将 $\Delta\sigma_\theta^2$ 作为对应于方位测量的 $\Delta\boldsymbol{R}(j, j)$元素的均值。

18. 原始[2×2]$\boldsymbol{R}$ 矩阵表示为 $\boldsymbol{R}_k = \boldsymbol{R}_{k-1} + \begin{bmatrix} \Delta\sigma_r^2 & 0 \\ 0 & \Delta\sigma_\theta^2 \end{bmatrix}$。

19. IF(在步骤 7 中观测到新特征)

THEN 根据式(7.20)~式(7.22)，计算增广状态矢量和误差协方差矩阵。

ENDIF

20. 返回步骤 1。

7.4 模糊神经网络架构及基于粒子群优化(PSO)算法的训练方法

7.4.1 模糊神经网络架构

所开发的模糊神经网络模型是一个单输入单输出的系统。四层架构如图 7.1 所示。设 u_i^l 和 O_i^l 分别表示第 l 层中第 i 个节点的输入和输出。

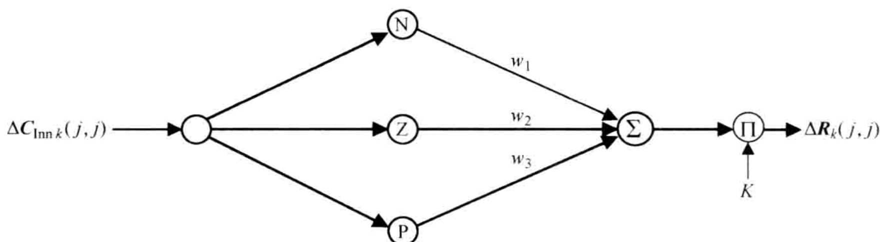


图 7.1 模糊神经网络系统的四层架构(经 IEEE 许可，从参考文献[44]中复制)

1. 第一层：输入层

该层只包括一个节点，表示单输入变量。该节点的输入、输出关系为

$$O^1 = u^1 = \Delta\,\hat{C}_{\mathrm{Inn}k}(j,\ j) \tag{7.30}$$

2. 第二层：隶属函数层

在该层中，输入变量通过三个隶属函数(MF)，即负(N)、零(Z)和正(P)进行模糊化处理。图 7.2 给出了这三个隶属函数，其中 N_{v} 和 N_{b} 分别表示隶属函数 N 的右顶点和右基点，Z_{bl}、Z_{vl}、Z_{vr}和 Z_{br}分别表示隶属函数 Z 的左基点、左顶点、右顶点和右基点，P_{b} 和 P_{v} 分别表示隶属函数 P 的左基点和左顶点。第 i 个隶属函数的输出为

$$O_i^2 = \mu_i(u^1) = \mu_i(\Delta\,\hat{C}_{\mathrm{Inn}k}(j,\ j)) \tag{7.31}$$

3. 第三层：去模糊化层

该层执行去模糊化操作，其中利用所有输入的加权平均来计算去模糊化后的输出。由此，该层中孤立节点的输出可计算为

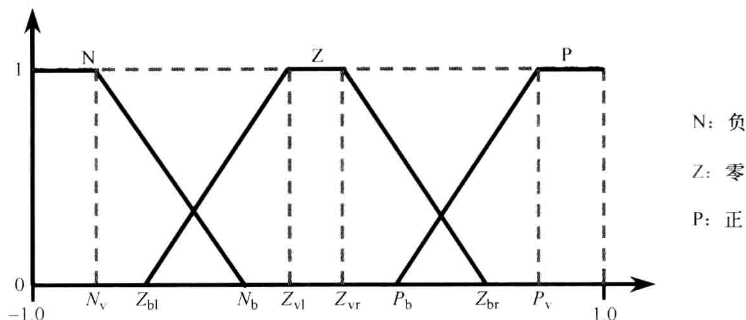


图 7.2　图 7.1 中所采用的隶属函数(经 IEEE 许可，从参考文献[44]中复制)

$$O^3 = \frac{\sum_{i=1}^{3} O_i^2 w_i}{\sum_{i=1}^{3} O_i^2} = \frac{\sum_{i=1}^{3} \mu_i(u^1) w_i}{\sum_{i=1}^{3} \mu_i(u^1)} \tag{7.32}$$

4. 第四层：输出层

该层对去模糊化后的输出进行适当缩放。该层中节点的输入、输出关系为

$$O^4 = Ku^4 = KO^3 \tag{7.33}$$

7.4.2　利用 PSO 训练模糊神经网络模型

通过训练该模糊神经网络模型来确定系统的自由参数，即隶属函数的参数、单个输出结果和输出增益。然而在传统的监督模式下，对于一个给定输入，由于期望输出无法定量确定而导致无法训练模型。因此，对于这种无监督模式的系统，不能采用常用的后向传播训练方式，而只能采用一种合适的随机全局优化算法。目前，已有多种可选的优化算法。本节将介绍为何 PSO 算法能够适用于这种情况。PSO 算法是一种基于鸟群或鱼群的群体行为的相对较新的算法[40,41]。模糊神经网络系统的训练可看作一个高维元启发式优化问题，目标是在变量值 x_1，x_2，…，x_n 的基础上来优化适应度函数 $f_{fit}(x_1, x_2, \cdots, x_n)$。

对于一个 PSO 问题，在一个多维空间中创建多个 x_1，x_2，…，x_n 的可能解(称为粒子)，并在每次迭代中估计每个粒子的适应性。在本书所考虑的情况下，每个粒子都由一个十二维矢量 $\boldsymbol{x} = [x_1, x_2, \cdots, x_{12}]^T$ 表示，如图 7.3 所示。每个粒子 i 是由表示在当前时间步长下的位置($\boldsymbol{x}_i$)和速度($\boldsymbol{v}_i$)矢量来表征的。为确定适应度函数(f_{fit})的最优值，在每个时间步长中调整每个粒子的速度($\boldsymbol{v}_i$)和位置($\boldsymbol{x}_i$)。每个时间步长中更新后的速度$\boldsymbol{v}_{inew}$是以下三个主元素的一个函数：同一粒子的原速度矢量($\boldsymbol{v}_{iold}$)、第 i 个粒子的最佳位置(记为 $\boldsymbol{p}_i$)与当前位置($\boldsymbol{x}_i$)(称为认知分量)之差，以及第 i 个粒子拓扑邻域中任一粒子的最佳位置(记为 $\boldsymbol{p}_g$)与第 i 个粒子当前位置($\boldsymbol{x}_i$)(称为“社会”分量)之差[40,41]。对后两个分量都随机加

权以使得在每个粒子速度更新时能产生足够振荡，从而能允许每个粒子在问题空间中寻找一种更佳的模式。因此，在第 d 维中，速度和位置的更新关系为

$$
\begin{aligned}
& v_{id\text{new}} = v_{id\text{old}} + \varphi_i(p_{id} - x_{id}) + \varphi_g(p_{gd} - x_{gd}) \\
& \text{IF}(v_{id\text{new}} > v_{d\max})\ \text{THEN}\ v_{id\text{new}} = v_{d\max}\ \text{ENDIF} \\
& \text{IF}(v_{id\text{new}} < -v_{d\max})\ \text{THEN}\ v_{id\text{new}} = -v_{d\max}\ \text{ENDIF} \\
& x_{id\text{new}} = x_{id\text{old}} + v_{id\text{new}} \\
& v_{id\text{old}} = v_{id\text{new}} \\
& x_{id\text{old}} = x_{id\text{new}}
\end{aligned}
\tag{7.34}
$$

其中，φ_i 和 φ_g 用于引入随机权重，分别记为 $\varphi_i = c_i \times \text{rand}_1(\,)$ 和 $\varphi_g = c_g \times \text{rand}_2(\,)$。$\text{rand}_1(\,)$ 和 $\text{rand}_2(\,)$ 为[0，1]范围内的两个随机函数，c_i 和 c_g 均为正常数。通常选择 c_i 和 c_g 为 $c_i = c_g = 2$。传统的 PSO 模型在开始阶段能够快速收敛，但经常会在确定最优解的微调阶段有问题。因此，本算法采用一种改进的 PSO 算法，即一种在迭代初期粗调，而在迭代后期微调的快速组合方法[40]。相应的速度更新规则为

$$
v_{id\text{new}} = w_{\text{iter}}(v_{id\text{old}}) + \varphi_i(p_{id} - x_{id}) + \varphi_g(p_{gd} - x_{gd}) \tag{7.35}
$$

而位置更新规则保持不变。w 称为惯性权重，该值在初始时较大，然后在迭代过程中逐步减小，从而使得其能够在速度更新中先粗调，然后逐步微调。在本算法中，采用了一个线性自适应的惯性权重，并在给定迭代中，w_{iter} 给出该惯性权重值。继续执行该迭代过程，直到在该优化过程中产生一个满意解。该解是根据是否 f_{fit} 值小于给定最大允许值或已达到最大迭代次数来评估的。参考文献[40，41]中对 PSO 算法进行了详细阐述。

N_v	N_b	Z_{bl}	Z_{vl}	Z_{vr}	Z_{br}	P_b	P_v	w_1	w_2	w_3	K

图 7.3 PSO 算法中每个十二维粒子的详细配置
（经 IEEE 许可，从参考文献[44]中复制）

在本方法中，模糊神经网络对基于 EKF 的 SLAM 的目标是尽可能地提高估计性能。这意味着在车辆/机器人运动过程中，应尽可能地使得在整个观测时刻集中新息序列的实际协方差和理想协方差之差最小。因此，可根据下列步骤来构造适应度函数：计算任意观测时刻时 $\boldsymbol{\Delta C}_{\text{Inn}k}$ 矩阵中所有对角元素的方均根；在当前迭代过程中，保存每个观测时刻的方均根；在一次完整迭代结束时，计算

所有观测时刻方均根的平均值。从数学上可表示为

$$f_{\text{fit}} = \frac{\sum_{n_{\text{obs}}=1}^{N_{\text{obs}}} \left(\frac{\sum_{j=1}^{J_{\text{C-nobs}}} [\Delta C_{\text{Inn}k}(j,j)]^2}{J_{\text{C-nobs}}} \right)}{N_{\text{obs}}} \tag{7.36}$$

式中 N_{obs}——在一次给定迭代中总的观测时刻数；

$J_{\text{C-nobs}}$——第 n_{obs} 次观测时，矩阵 $\Delta C_{\text{Inn}k}$ 中对角元素的总个数。

对于一种有实际意义的自适应 NFS，每次迭代结束时每个粒子的位置都会受到若干约束条件的限制。这些约束大多数是为了保证隶属函数的特定形状(通常为梯形，某些特殊情况下也可以是矩形)，并保证连续隶属函数的延长线重叠。另一类约束是应以单调非降的形式来选择每个隶属函数的控制点(从左到右开始)。这样就能确保在 NFS 输入范畴中的所有区域将包含至少一个隶属函数。上述约束条件具体表现如下：

IF($N_b < N_v$) THEN $N_b = N_v$ ENDIF
IF ($z_{vl} < z_{bl}$) THEN $z_{vl} = z_{bl}$ ENDIF
IF ($z_{vr} < z_{vl}$) THEN $z_{vr} = z_{vl}$ ENDIF
IF ($z_{br} < z_{vr}$) THEN $z_{br} = z_{vr}$ ENDIF
IF ($P_v < P_b$) THEN $P_v = P_b$ ENDIF
IF ($N_b < Z_{bl}$) THEN $N_b = Z_{bl}$ ENDIF
IF ($Z_{br} < P_b$) THEN $Z_{br} = P_b$ ENDIF (7.37)

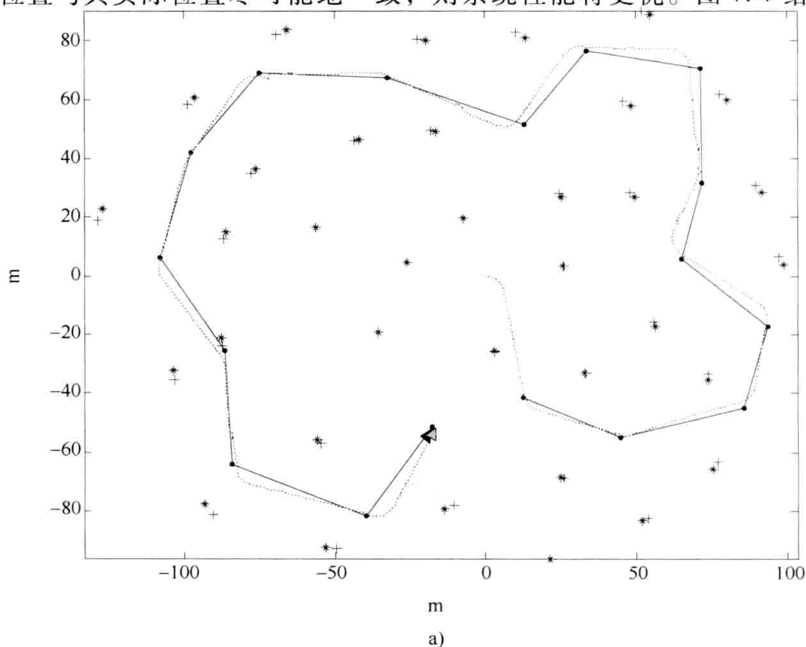
另一类约束是表示对 NFS 输出层的缩放只是幅值上的比例缩放。因此，该约束不能用于改变输出极性，这也意味着 K 不能为负值。

7.4.3 性能评估

为了评估本书系统的性能，在此考虑参考文献[42]中所述的不同环境。实际上，参考文献[42]所提供的软件包可作为学习和分析现有基于 EKF 和粒子滤波的 SLAM 算法的一个良好平台。在该平台上，研究人员可开发各自的算法并进行性能比较。在此，提供了几种基准环境，并在这些仿真环境下结合所关联的给定车辆/移动机器人运动模型来测试本书所提出的算法。通常以一种方式来定义环境，即假设通过某些路标点来进行车辆/机器人导航，并在此过程中，能够获得包含一些特征/路标点的环境地图。在所提的方法中，考虑了参考文献[42] 中所指定的三种环境。每种环境下，都认为理想机器人在由十七个导航点确定的路径上运动。然而，每种环境中所包含的路标点个数会变化以增加复杂度，在此所考虑的三种环境中分别包含 35、135 和 497 个路标。指定输入控制量中的不确定性为 $\sigma_w = 0.3\text{m/s}$，$\sigma_s = 3°$。经过八个连续的预测步骤之后，执行一

次观测和相关联的更新，这与参考文献[42]中基于 EKF 的算法相同。在此，沿用基于 EKF 的 SLAM 算法中的常用概念，并不是在每次预测之后都执行一次观测和更新，而是计算几个连续预测之后，由一次观测和更新来修正。这样有助于减少 SLAM 算法的计算量。在算法 7.1 中，用 Time_ for_ Observation 标志位来标识，即每八次连续迭代之后，下一次迭代设为真。

在此，对本书所提出的系统与传统的 **Q**、**R** 矩阵在整个实验过程中保持不变的基于 EKF 的 SLAM 系统进行了性能比较。在本书算法中，也是以相同的 **Q**、**R**、矩阵开始执行，但不同的是，不断自适应 **R** 矩阵。根据参考文献[42]所提供的数据，当已知传感器统计值为 $\sigma_r=0.1\text{m}$，$\sigma_\theta=1°$时，基于 EKF 的 SLAM 算法效果很好。首先，考虑若传感器统计值被误认为 $\sigma_r=0.2\text{m}$，$\sigma_\theta=0.1°$时的情况。后文介绍的图中，细实线表示机器人实际经过的路径，而点线表示根据每次采样时刻或迭代过程中机器人位姿的估计状态由 SLAM 算法估计的路径，星形线(*)表示环境中静态路标的实际位置，十字形线(+)表示测试结束后这些路标的估计位置。显然，如果机器人的估计路径和实际路径尽可能地匹配，且路标的估计位置与其实际位置尽可能地一致，则系统性能将更优。图 7.4 给出了三



a)

图7.4 示例 I 中($\sigma_r=2.0\text{m}$，$\sigma_b=0.1°$)[①]传统的基于 EKF 的 SLAM 算法的性能，其中环境中具有 35(见图 7.4a)、135(见图 7.4b)和 497(见图 7.4c)个特征/路标(经 IEEE 许可，从参考文献[44]中复制)

① 此处原书有误，σ_b 应为 σ_θ。——译者注

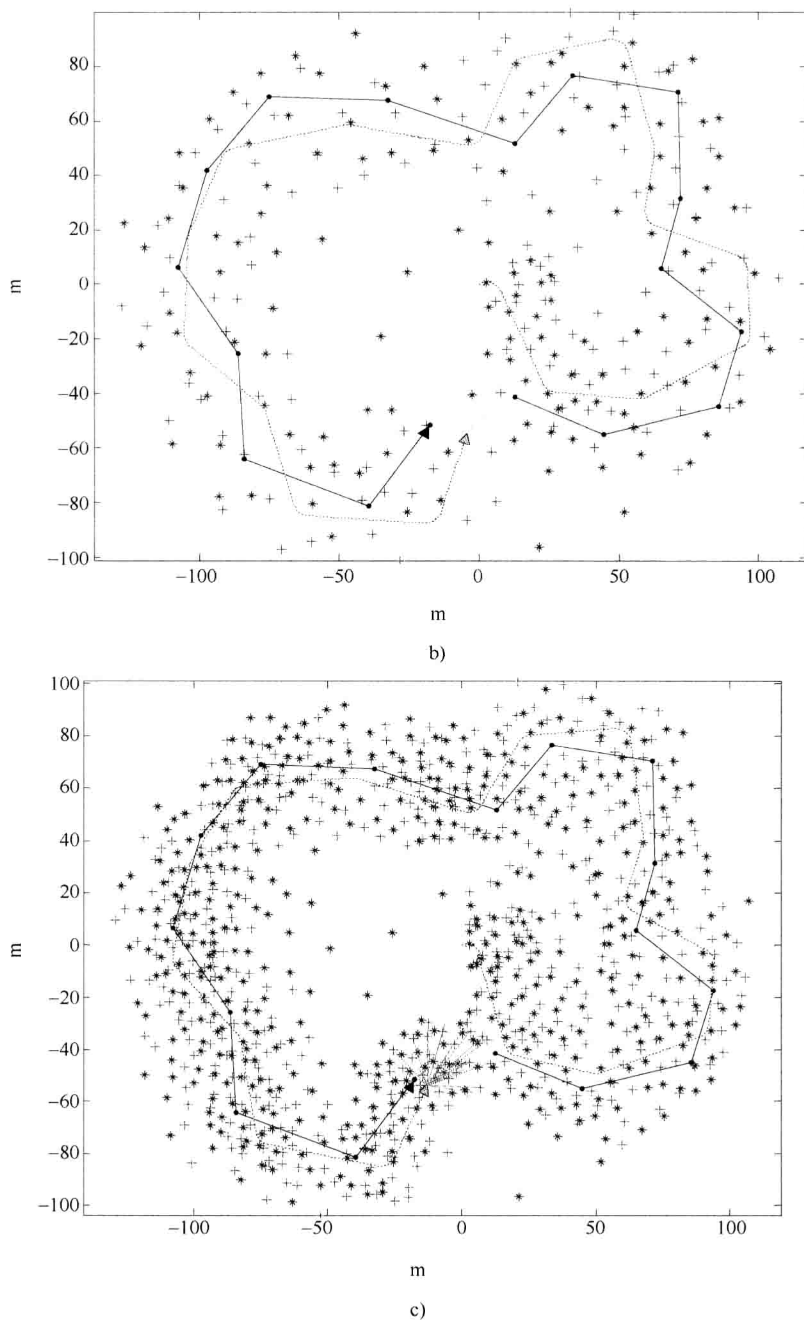


图 7.4 示例 I 中($\sigma_r=2.0\text{m}$，$\sigma_b=0.1°$)传统的基于 EKF 的 SLAM 算法的性能，其中环境中具有 35(见图 7.4a)、135(见图 7.4b)和 497(见图 7.4c)个特征/路标(经 IEEE 许可，从参考文献[44]中复制)(续)

种不同环境情况下传统的基于 EKF 的 SLAM 算法的性能。由图可知，若环境中的路标较少，系统性能还是可接受的。然而当路标变得更加密集时，系统性能就会变得相当差，并且随着 EKF 的估计位置与机器人实际位置相距甚远，则不同采样时刻机器人的估计位置和所构建的地图也将急剧变差。图 7.5 给出了相同环境下采用结合模糊神经网络的基于 EKF 的 SLAM 算法的情况。由图可知，模糊神经网络能够在很大程度上提高系统性能，同时在路标个数变化的三种不同环境下，机器人的状态估计以及所构建的地图也都很稳定。在所有三种环境中，机器人的估计位置与其实际位置非常接近，静态路标的估计位置也与其实际位置非常匹配。

进一步对传感器统计值被误认为相反方向（即 $\sigma_r=0.01\mathrm{m}$，$\sigma_\theta=3.0\mathrm{deg}$）的另一种情况进行测试。在此，对同样的环境集采用了相同的算法集。图 7.6 给出了传统基于 EKF 的 SLAM 算法的性能，图 7.7 给出了结合模糊神经网络的基于 EKF 的 SLAM 算法的性能。在这些示例情况下，基于 EKF 的 SLAM 算法表现出不同的性能趋势。

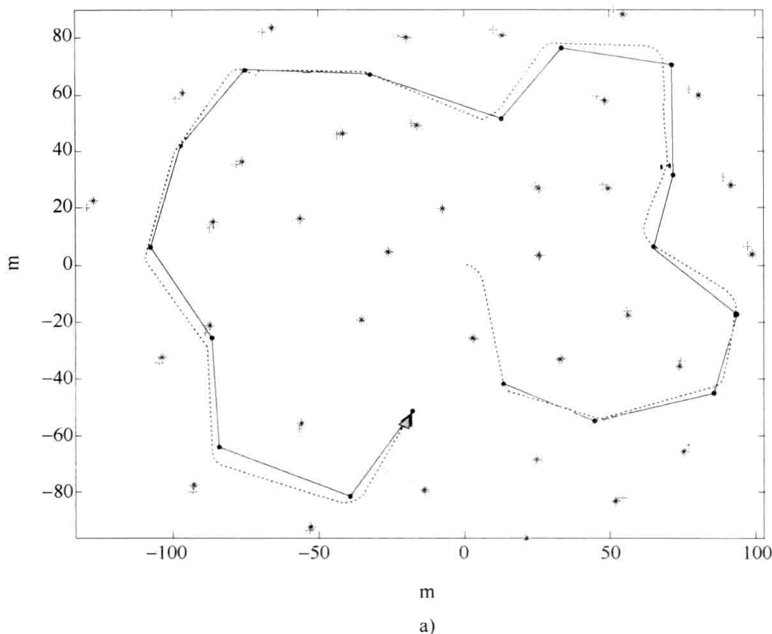
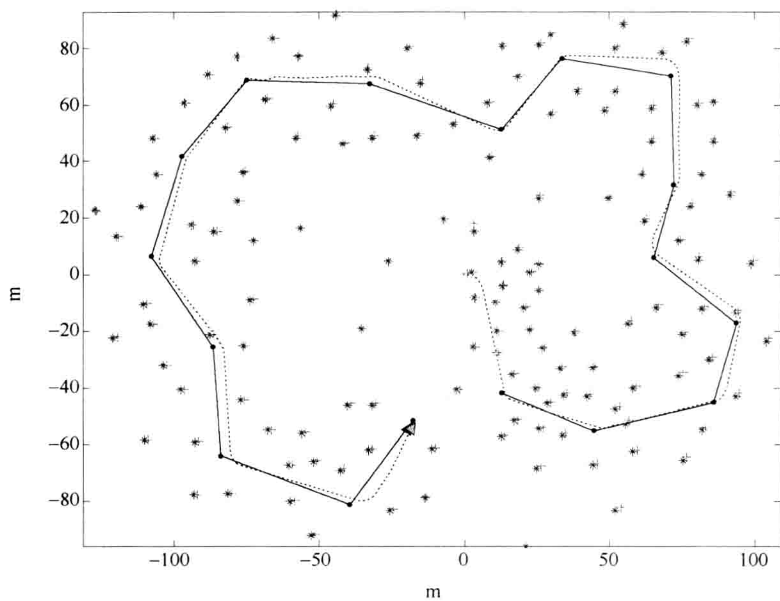
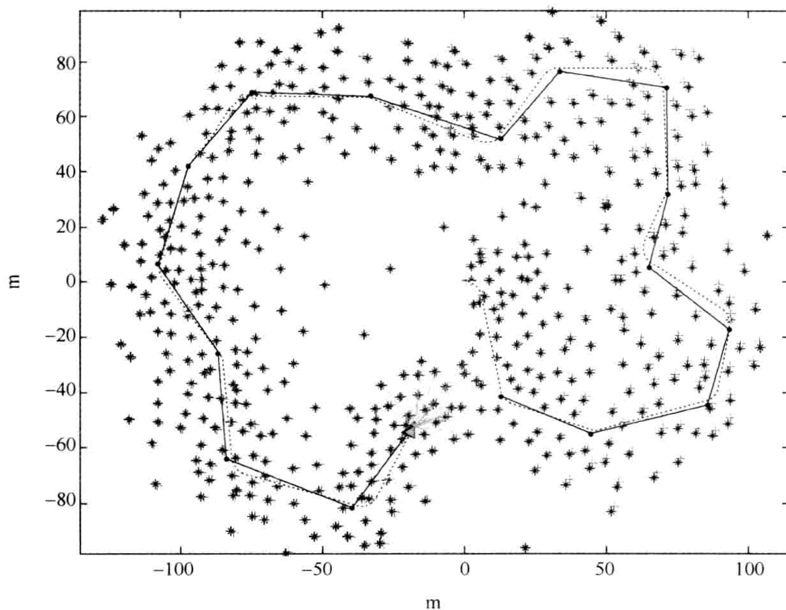


图 7.5　示例 I 中（$\sigma_r=2.0\mathrm{m}$，$\sigma_b=0.1°$）[①]结合模糊神经网络的基于 EKF 的 SLAM 算法的性能，其中环境中具有 35（见图 7.5a）、135（见图 7.5b）和 497（见图 7.5c）个特征/路标（经 IEEE 许可，从参考文献[44]中复制）

① 此处原书有误，σ_b 应为 σ_θ。——译者注



b)



c)

图 7.5 示例 I 中($\sigma_r = 2.0\text{m}$，$\sigma_b = 0.1°$)结合模糊神经网络的基于 EKF 的 SLAM 算法的性能，其中环境中具有 35(见图 7.5a)、135(见图 7.5b) 和 497 (见图 7.5c)个特征/路标（经 IEEE 许可，从参考文献[44]中复制)(续)

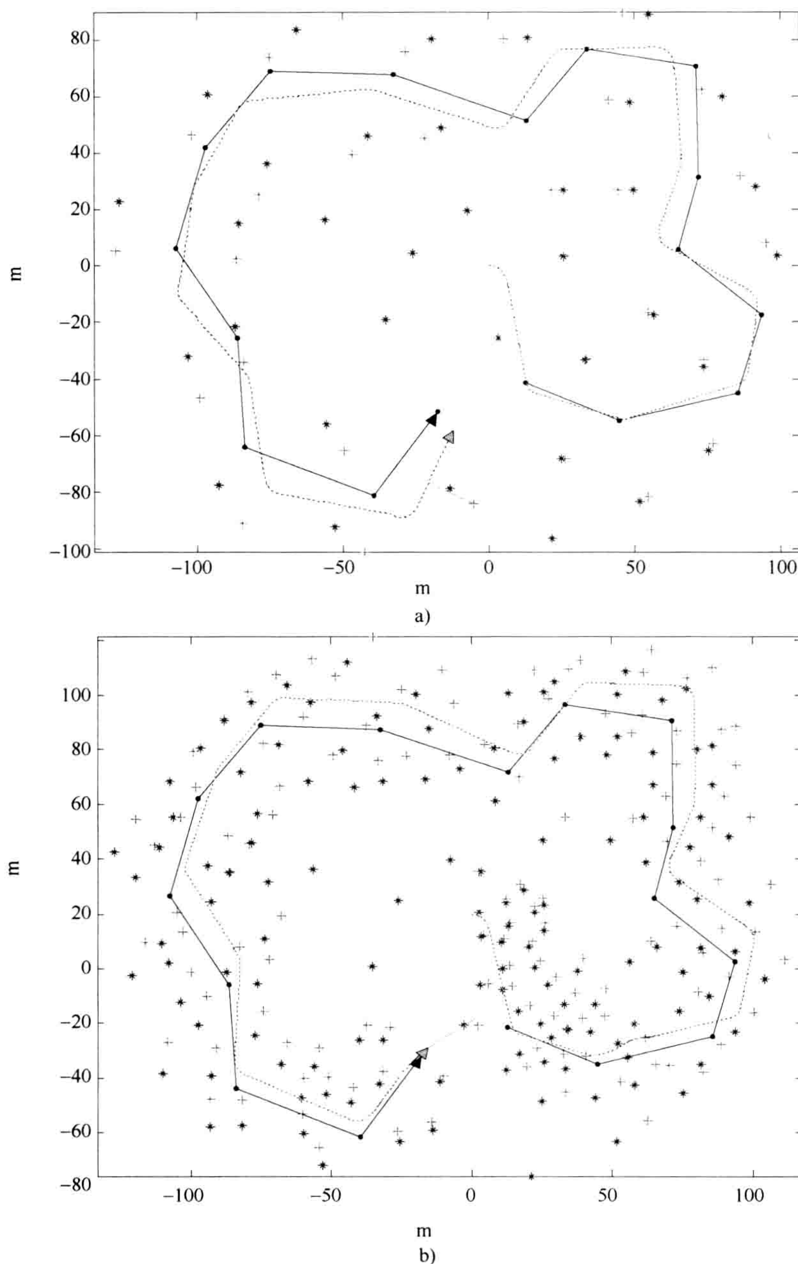
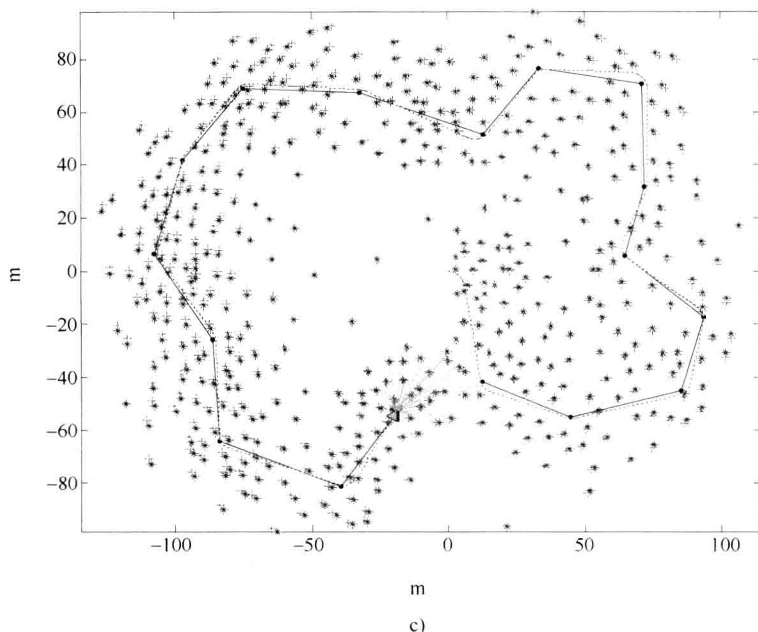


图 7.6　示例Ⅱ中($\sigma_r=0.01\text{m}$，$\sigma_b=3.0°$)[①]传统的基于 EKF SLAM 性能，其中环境中具有 35(见图 7.6a) 135(见图 7.6b)和 497(见图 7.6c)个特征/路标（经 IEEE 许可，从参考文献[44]中复制）

① 此处原书有误，σ_b 应为 σ_θ。——译者注



c)

图 7.6 示例Ⅱ中($\sigma_r=0.01\text{m}$，$\sigma_b=3.0°$)传统的基于 EKF SLAM 性能，其中环境中具有35(见图 7.6a) 135(见图 7.6b)和 497(见图 7.6c)个特征/路标（经 IEEE 许可，从参考文献[44]中复制)(续)

由图可看出，环境中包含少量路标时估计性能最差。然而随着路标个数增加，估计得也越来越准确。对于包含 497 个路标的情况，基于 EKF 的 SLAM 算法性能已经相当满意了。另一方面，结合模糊神经网络的基于 EKF 的 SLAM 算法对于每种环境都表现非常稳定，机器人位姿和特征位置都估计得相当准确。如图 7.5 和图 7.7 所示，结合模糊神经网络的基于 EKF 的 SLAM 算法的每个结果都是在一次运行中得到的。对于两种示例情况下的 6 个特定场景，共进行了 10 次独立的运行。实验发现，对于每种特定情况，10 次独立运行的结果都非常接近。同时，这些实验也进一步证明了在传感器统计值有误差的情况下，结合模糊神经网络的基于 EKF 的 SLAM 算法可极大地提高传统算法的性能。在上述情况下，传统 EKF 算法的性能相当不可靠。然而模糊神经网络能够帮助 EKF 算法保持一个稳定的性能，且已表明在传感器统计值有误差的几种环境下，该性能具有足够的鲁棒性。

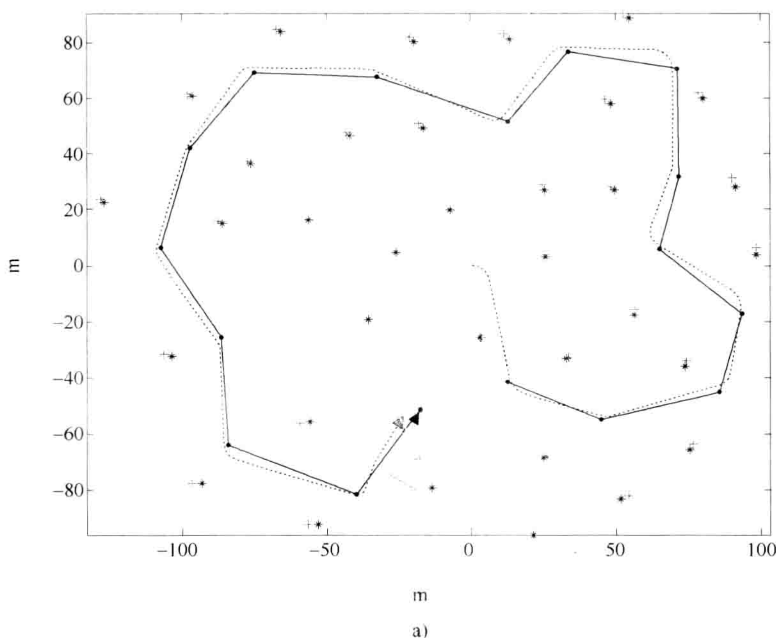
对于结合模糊神经网络的基于 EKF 的 SLAM 算法，在上述讨论的每种情况下，都能根据一个给定的环境下机器人所采集的数据来离线训练模糊神经网络系统。在实验中，也实现了包含 135 个路标的每种示例环境下的训练过程。一旦模糊神经网络系统训练完成(在一个给定路标配置的基础上)，并确定 NFS 的自由参数之后，就针对如上所述的几种路标配置(即环境中包含 35、135 和 497

个路标)，利用训练后的基于 NFS 的 EKF 算法来实现机器人导航。表 7.1 详细列出了训练 NFS 时所用 PSO 算法的参数。在此，用于学习 NFS 中隶属函数控制点(即$[x_1x_2\cdots x_8]$)的每个粒子的维数均初始化为[-1, 1]范围内的粒子位置，这是在与 NFS 共同工作的归一化过程中构成的。与 NFS 中第三层关联的预期权重(由 PSO 算法的维数 x_9、x_{10}和 x_{11} 表示)均初始化为[-2, 2]范围内的位置，与 NFS 中第四层关联的预期增益 K(由 PSO 算法的维数 x_{12}表示)初始化为[0, 2]范围内的位置，这是因为在此假设 K 为非负值。PSO 算法的终止条件每次都设为最大迭代次数 20。初始传感器信息 $\sigma_r=0.2\text{m}$，$\sigma_\theta=0.1°$ 的示例情况下，训练过程完成时 NFS 的学习参数为

$$[x_1x_2\cdots x_{12}] = [-0.2008 \quad -0.0626 \quad -0.0626 \quad -0.0626 \quad 0.0820 \quad 0.5961 \quad 0.3224 \quad 0.4002 \quad -0.0086 \quad 1.5801 \quad -0.9729 \quad 0.0011]$$

对于初始传感器信息 $\sigma_r=0.01\text{m}$，$\sigma_\theta=3.0°$ 的示例情况，NFS 的学习参数为

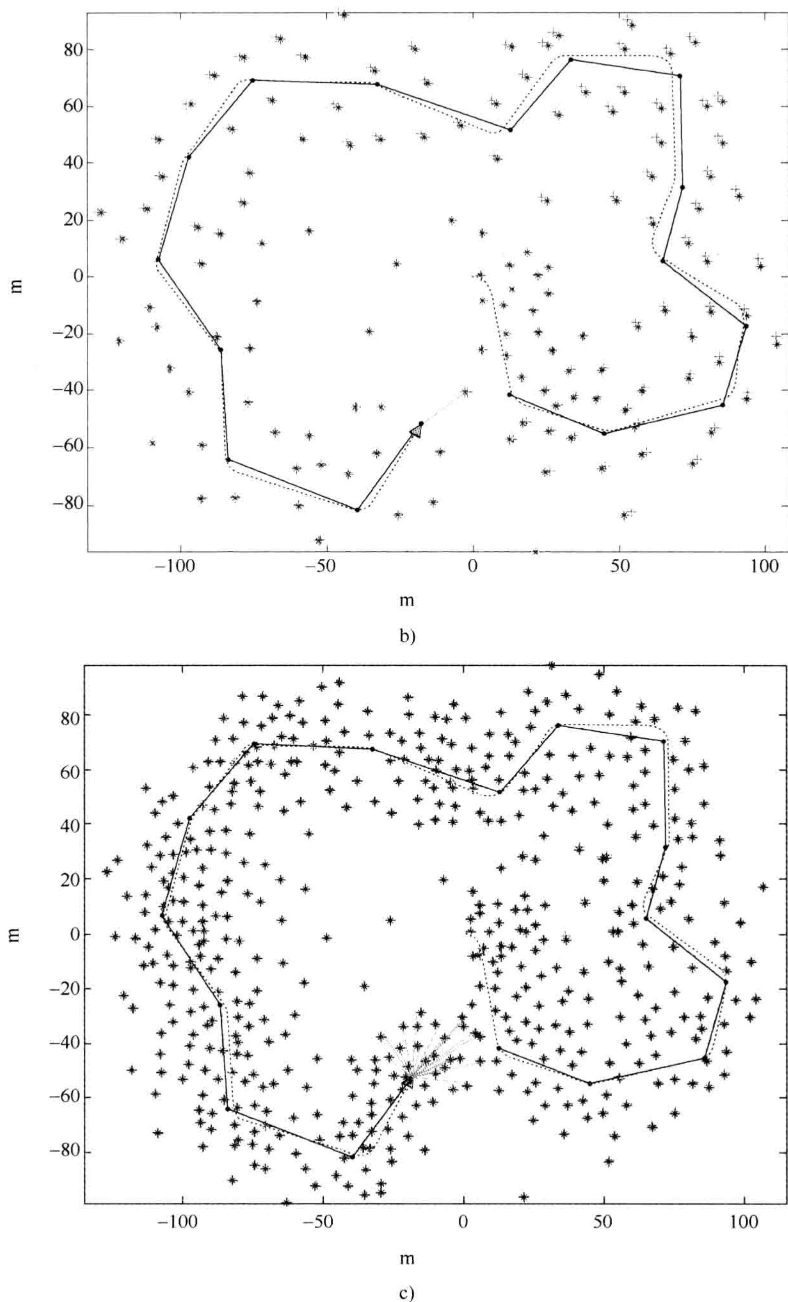
$$[x_1x_2\cdots x_{12}] = [-0.4570 \quad 0.5242 \quad 0.4805 \quad 0.9741 \quad 0.9741 \quad 0.9741 \quad -0.4290 \quad 0.2413 \quad -0.0024 \quad -0.8762 \quad 1.3561 \quad 0.2907]$$



a)

图 7.7　示例Ⅱ中($\sigma_r=0.01\text{m}$，$\sigma_b=3.0°$)[①]结合模糊神经网络的基于 EKF 的 SLAM 算法的性能，其中环境中具有 35(见图 7.7a)、135(见图 7.7b) 和 497(见图 7.7c)个特征/路标（经 IEEE 许可，从参考文献[44]中复制）

① 此处原书有误，σ_b 应为 σ_θ。——译者注



b)

c)

图 7.7　示例Ⅱ中($\sigma_r=0.01\mathrm{m}$，$\sigma_b=3.0°$)结合模糊神经网络的基于 EKF 的 SLAM 算法的性能，其中环境中具有 35(见图 7.7a)、135(见图 7.7b) 和 497(见图 7.7c)个特征/路标（经 IEEE 许可，从参考文献[44]中复制)(续)

由上可看出，在每种情况下，这些学习参数都满足式(7.37)中的约束条件。

表 7.1 所采用的 PSO 算法参数

SI 序号	参数描述	示例 I 的参数值	示例 II 的参数值
1	粒子数(N)	40	40
2	维数(D)	12	12
3	初始惯性权重($W_{initial}$)	0.9	0.9
4	惯性权重(ΔW)	2.5e-4	2.5e-4
5	隶属函数的初始范围(x_1, x_2, …, x_8)	[-1, 1]	[-1, 1]
6	权重因子的初始范围(x_9, x_{10}, x_{11})	[-2, 2]	[-2, 2]
7	增益的初始范围(x_{12})	[0, 2]	[0, 2]
8	隶属函数的最大允许速度(v_{1max}, v_{2max}, …, v_{8max})	0.3	0.1
9	权重因子的最大允许速度(v_{9max}, v_{10max}, v_{11max})	1.0	0.5
10	增益的最大允许速度(v_{12max})	1.0	0.5

7.5 采用 DE 优化算法训练模糊监督器

7.4 节阐述了如何利用 PSO 算法训练一个模糊/模糊神经网络监督器来成功地监督基于 EKF 的 SLAM 系统。从逻辑上来说，该思想可扩展到其他进化算法来实现类似的基于模糊/模糊神经网络的监督。因此，在此实现了一种采用差分进化(DE)优化算法的类似的模糊监督器，差分进化算法是另一种解决该类问题的常用进化算法[45]。类似于其他常用的全局优化方法，差分进化算法也是在多维搜索空间中同时创建几个候选解，其中每个解都包含一个该优化问题的可能解矢量，并且，每个解都由表示特定解进化为最优解的适应性程度的适应度函数来单独评估。迭代执行该过程，其中从之前所产生的可能解中创建新矢量(即可能的候选解)，来产生可由适应度函数值定量估计的越来越优的解。根据上次产生的候选矢量，可采用几种数学策略来创建本次新的候选矢量。在每次产生结束时，候选解能够给出可能产生最优解的最佳适应度函数值(通常为最小值)。继续该迭代过程，直到某次产生最优解向量的最佳适应度函数值低于该优化过程所允许的最大适应度函数值，或达到最大进化代数。

此处，考虑在 D 维 $\boldsymbol{x}$ 的基础上，用于最小化代价函数 $f(\boldsymbol{x})$ 的基本差分进化算法中，(x_1, x_2, …, x_D)的候选解的 NP 个数创建在一个 D 维空间中，并在每代 G 中评估每个解的适应性。随机产生初始种群，目标是使得产生的矢量应尽可能地涵盖整个搜索空间。通过在第三个矢量中增加两个种群矢量之间的权重

之差(这三个矢量都属于第 G 代)来创建第 $G+1$ 代中所有第 i 个矢量，这可由下式表示[46,47]：

$$v_{i,G+1} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G}) \tag{7.38}$$

式中 $i=1, 2, \cdots, \mathrm{NP}$；

$r_1, r_2, r_3 \in [1, \mathrm{NP}]$且均相互不同；

F——一个恒定的权重因子，且通常 $F\in[0, 2]$，该因子可影响差值($x_{r_2,G} - x_{r_3,G}$)的幅值。

为增加新产生矢量的多样性，在此引入交叉方法。交叉操作可从新产生的扰动矢量 $\boldsymbol{v}_{i,G+1}$和原矢量 $\boldsymbol{x}_{i,G}$中产生一个新矢量 $\boldsymbol{u}_{i,G+1}$。在基本差分进化算法中，所产生的该新矢量为[11,12]：

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \cdots, u_{Di,G+1})$$

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{对于} \quad j = \langle n+1 \rangle_D \cdots \langle n+L \rangle_D \\ x_{ji,G} & \text{对于} \quad \text{其他}\, j \in [1, D] \end{cases} \tag{7.39}$$

式中 n——一个随机选取的整数，$n\in[1, D]$，并且决定了交叉操作的起始序号。

在这种基本的差分进化算法中，交叉长度或持续时间是根据所选择的交叉概率($\mathrm{CR}\in[0, 1]$)在$[1, D]$中选取的一个整数。对每个 $\boldsymbol{u}_{i,G+1}$都重新选择这些 n 和 L 值。

如果新矢量 $\boldsymbol{u}_{i,G+1}$可产生一个较小的适应度函数值，则该矢量就是第 $G+1$ 代的新 $\boldsymbol{x}_{i,G+1}$，否则保持 $\boldsymbol{x}_{i,G+1}=\boldsymbol{x}_{i,G}$。

7.5.1 性能评估

通过创建一个仿真环境，并采用参考文献[42]中所提供的软件包，按照前面示例中的设置，对基于差分进化优化算法的模糊监督器求解 SLAM 问题的性能进行测试。在新的示例中，考虑一个不同环境下两组传感器统计值信息有误的情况：$\sigma_r=0.01\mathrm{m}$，$\sigma_b=10.0°$[㊀]；$\sigma_r=0.01\mathrm{m}$，$\sigma_b=15.0°$。针对这些情况，图7.8 给出了传统的基于 EKF 的 SLAM 算法[42]的性能结果。由图可知，所估计的机器人路径与理想路径存在很大偏差，并且所估计的许多路标位置也与其实际位置相差甚远。然而若对两种示例采用基于差分进化优化的模糊监督器的系统，模糊监督都能大幅提高性能，如图 7.9 所示。对于模糊监督算法，估计的机器人路径与理想路径偏差较小。在该方法中，以 $D=11$ 的差分进化并结合二项式交叉来自动学习模糊监督器的自由参数。所采用的差分进化算法是一种称为

㊀ 本节中的 σ_b 原书有误，均应为 σ_θ。——译者注

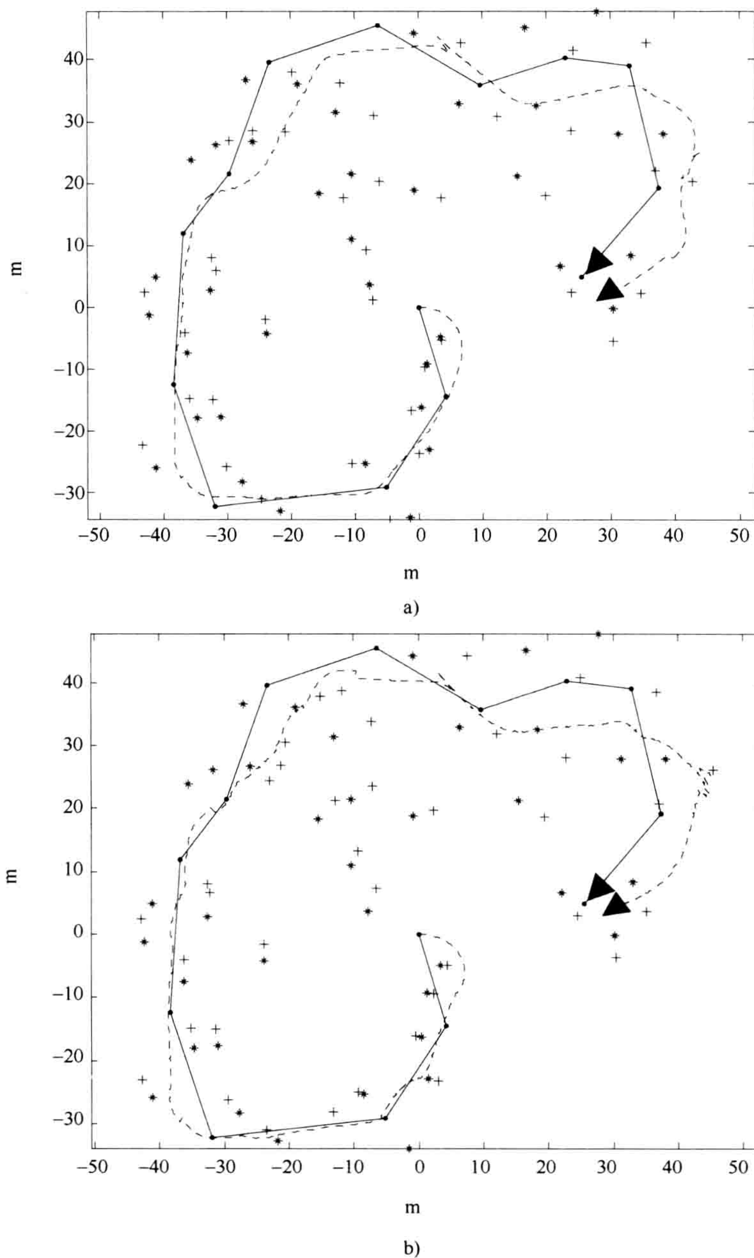


图 7.8　在传感器统计信息有误的情况下，传统的基于 EKF 的 SLAM 算法的性能

a) $\sigma_r = 0.01\text{m}$，$\sigma_b = 10.0°$　b) $\sigma_r = 0.01\text{m}$，$\sigma_b = 15.0°$

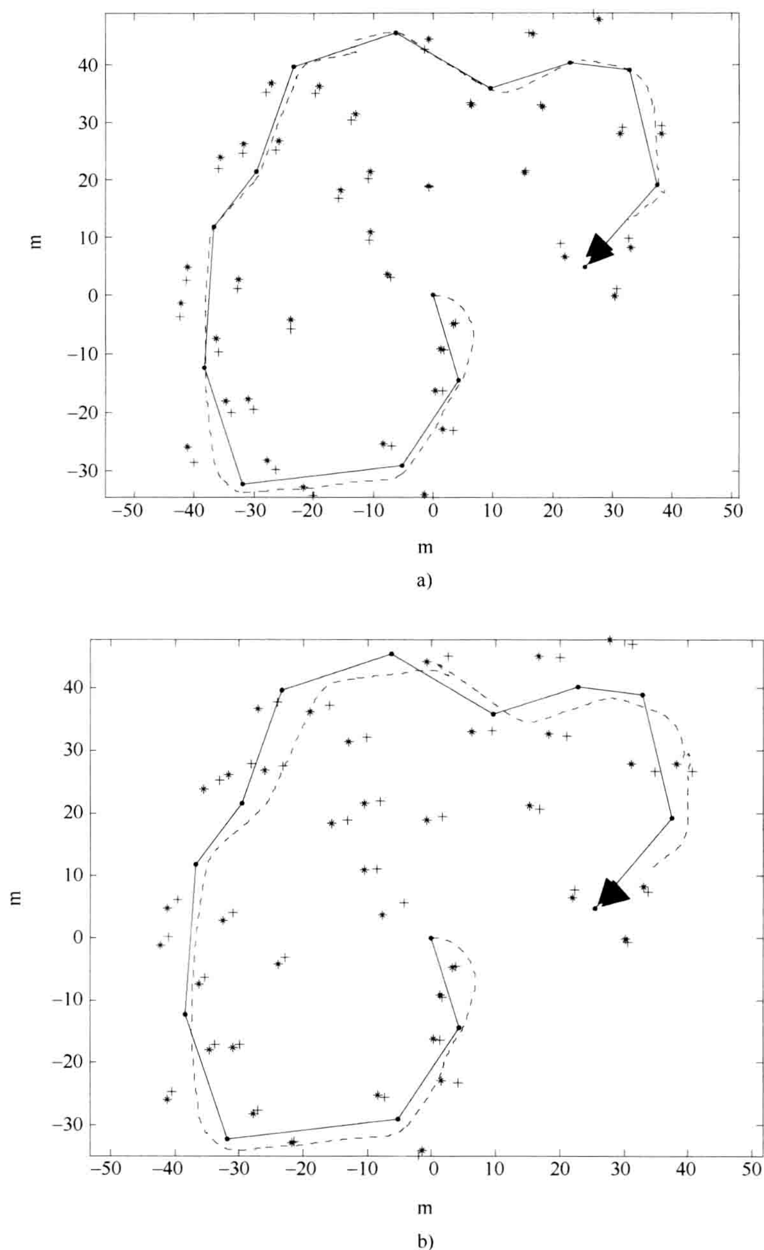


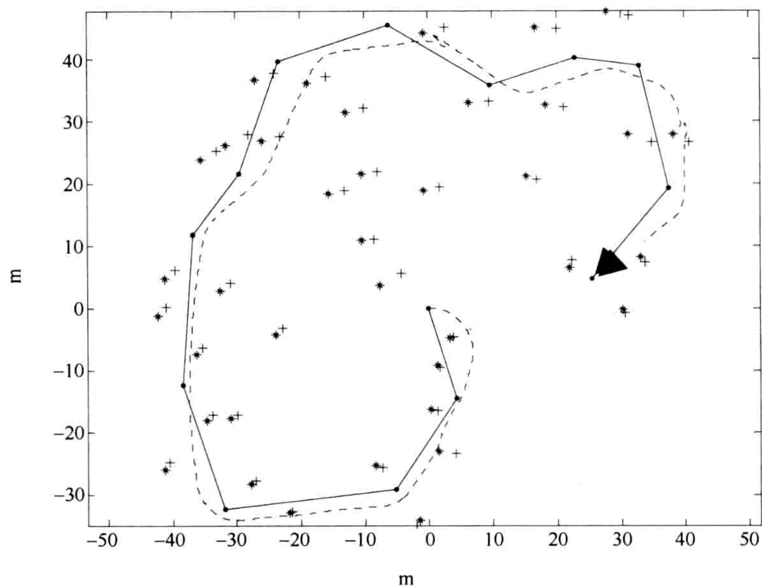
图 7.9 在传感器统计信息有误的情况下，结合模糊监督的传统的基于 EKF 的 SLAM 算法的性能

a) $\sigma_r = 0.01\mathrm{m}$，$\sigma_b = 10.0°$ b) $\sigma_r = 0.01\mathrm{m}$，$\sigma_b = 15.0°$

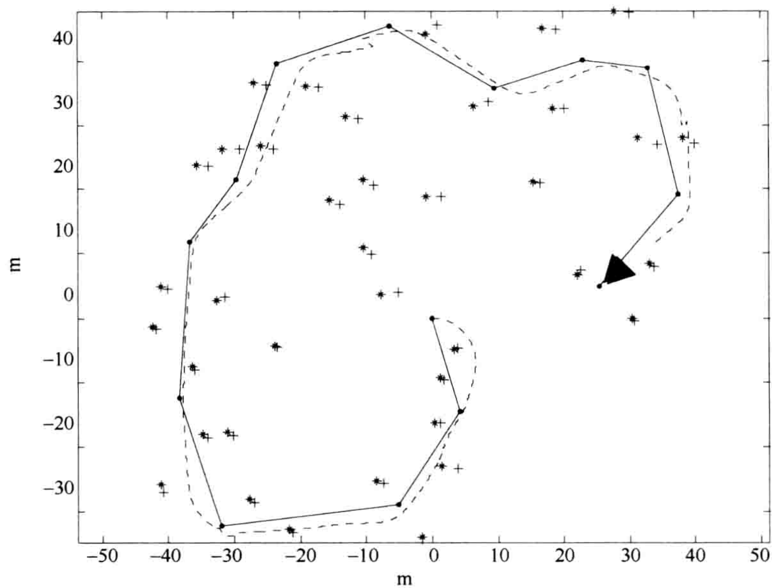
“DE/rand/1”的常用形式[46,47]。然而这种算法与原始的“DE/rand/1”方法有很大不同，由于是在对包含种群的数组重新排列后随机选取矢量的，因此一个给定矢量不能在含有扰动的同一项中重复选择两次[48]。另外，还可看出，每种示例下，路标的估计位置与其实际位置都要比采用传统的基于 EKF 的 SLAM 算法更加接近。

利用差分进化算法训练的模糊监督器，并选择控制参数为 NP = 20、$F=0.1$ 和 CR = 0.5，所得到的结果如图 7.9 所示。与大多数其他的随机全局优化方法相同，差分进化策略的性能会根据选择不同的自由参数而发生很大变化。因此选择合适的或准确拟合这些自由参数非常关键。根据参考文献[46]中提出的一般准则，对于许多应用，认为选择 NP = 10D，$F\in[0.5,\ 1]$，CR $\in[0,\ 1]$且远小于 1 是比较理想的。在这些参数中，F 是最关键的控制参数，其次是 NP 和 CR。因此，为达到差分进化的最佳性能，应采用各种不同的控制参数值来进行仿真，并观察相应的性能。对于传感器统计值为 $\sigma_r=0.01\mathrm{m}$，$\sigma_b=10.0°$ 的示例情况。首先，NP 和 CR 分别保持为 20 和 0.5，F 在 0 ~ 1 变化，并在每种情况下对模糊监督器进行训练。尽管根据一般准则，NP 应选择为 10 × 11 = 110，但这仍将会大大增加训练过程的计算量。因此，考虑到将计算量维持在较小的一个合理范围内，优化过程的 NP 值选为 20。在此，随着 F 值的变化，表明若采用较小的 F 值，则整个系统的性能会越来越好。最后得出在 $F=0.1$ 时可达到最佳性能，若 F 值更小，性能会有所下降，而 F 值较大时，性能下降明显。图 7.10 给出了 $F=0.05$、$F=0.1$ 和 $F=0.5$ 时，具有训练模糊监督器的系统的各自性能。图 7.11 给出了 F 为五种典型值下，随着机器人增量运动，每个采样时刻估计$\hat{\boldsymbol{x}}$的方均根误差。由此可得出 $F=0.1$ 时训练过程可产生最佳的实验结果。

根据该 F 值，可继续确定 NP 和 CR 的最佳值。保持 $F=0.1$ 和 CR = 0.5，变化 NP 值，目的是在 NP 尽量小的情况下达到一个满意性能，从而使得计算量最小。图 7.12 给出了 NP = 15、20 和 25 时机器人增量运动下，每个采样时刻所估计的$\hat{\boldsymbol{x}}$的方均根误差。由图可知，NP = 20 时可获得最佳性能，如果增大或减小 NP 值都会使得性能下降，因此在训练过程中选取 NP = 20。接下来，保持 $F=0.1$ 和 NP = 20，变化 CR 值。在此发现 CR 的变化并不是改变训练性能的关键因素。图 7.13 给出了 CR = 0.4、0.5 和 0.6 时$\hat{\boldsymbol{x}}$估计的方均根误差。尽管对于其他 CR 值性能都基本上非常相近，但还是可看出 CR = 0.5 时能够达到最佳性能。因此，可得出模糊监督器训练过程中差分进化算法的最佳控制参数应是 NP = 20、$F=0.1$ 和 CR = 0.5。对于每种传感器统计的示例情况下，采用这些参数对模糊监督器进行训练，即 $\sigma_r=0.01\mathrm{m}$，$\sigma_b=10.0°$；$\sigma_r=0.01\mathrm{m}$，$\sigma_b=15.0°$。图 7.9 给出了这些示例的性能结果。



a)



b)

图 7.10 以 NP = 20、CR = 0.5 进行基于差分进化算法训练的模糊监督的基于 EKF 的 SLAM 算法性能

a) $F = 0.05$ b) $F = 0.1$

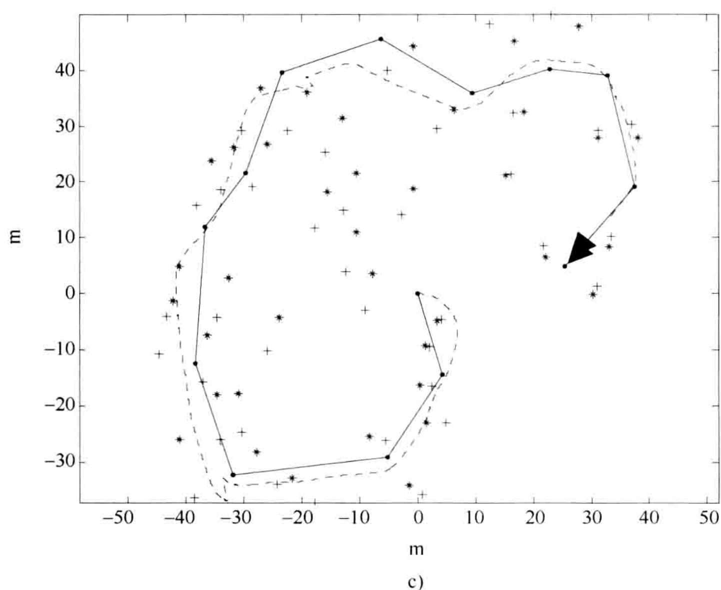


图 7.10 以 NP = 20、CR = 0.5 进行基于差分进化算法训练的模糊监督的基于 EKF 的 SLAM 算法性能(续)

c) $F=0.5$

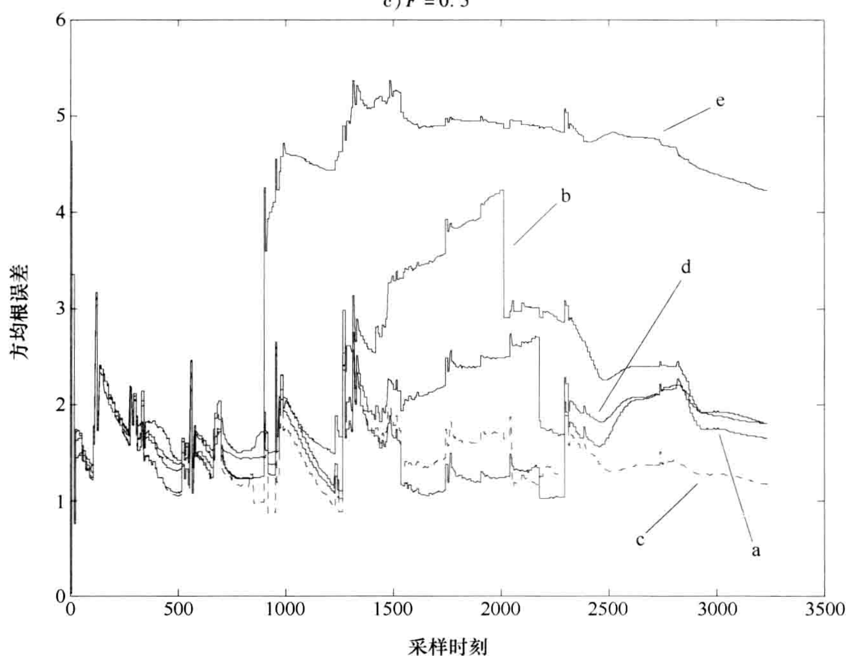


图 7.11 以 NP = 20、CR = 0.5 进行基于差分进化算法训练的模糊监督的基于 EKF 的 SLAM 算法性能

a—$F=0.05$ b—$F=0.08$ c—$F=0.1$ d—$F=0.15$ e—$F=0.5$

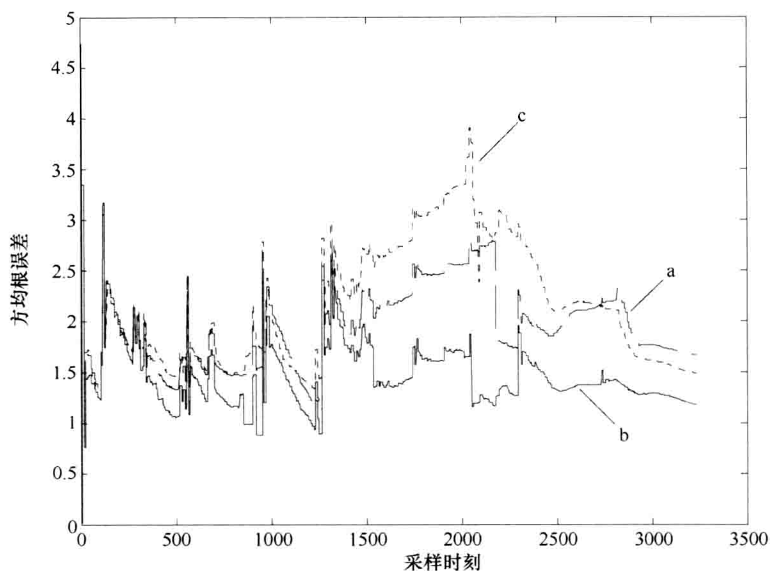


图 7.12 以 $F=0.1$、$CR=0.5$ 进行基于差分进化算法训练的模糊监督的基于 EKF 的 SLAM 算法性能

a—NP = 15 b—NP = 20 c—NP = 25

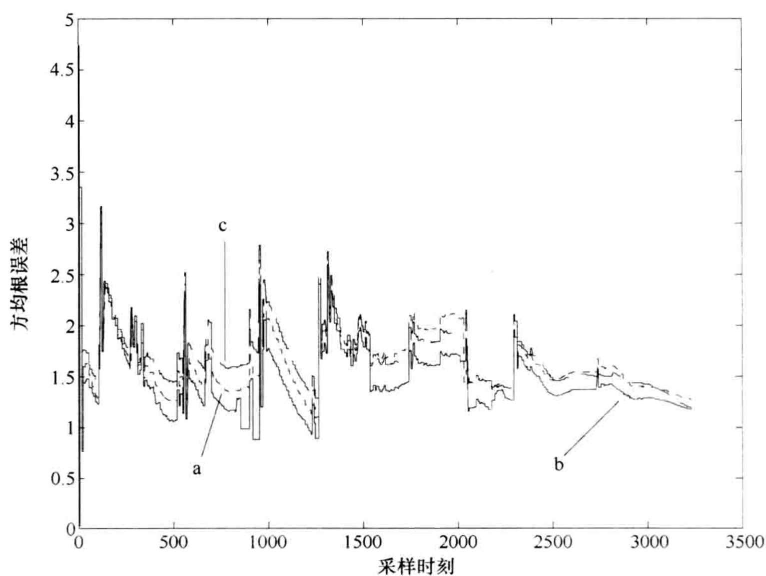


图 7.13 以 $F=0.1$、$NP=20$ 进行基于差分进化算法训练的模糊监督的基于 EKF 的 SLAM 算法性能

a—CR = 0.4 b—CR = 0.5 c—CR = 0.6

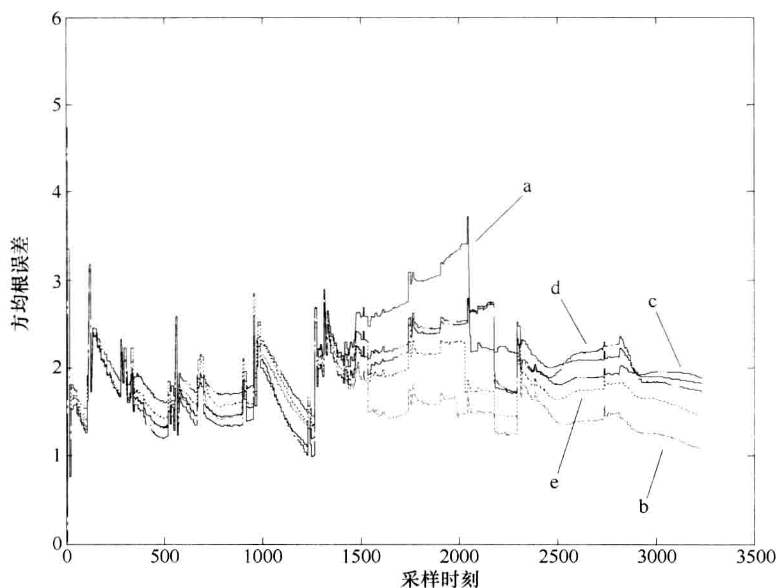


图 7.14　以惯性权重斜率为 2.0e－4a)、2.5e－4b)、5.0e－4c)、4e－2d)和 5e－2e)进行基于 PSO 算法训练的模糊监督的基于 EKF 的 SLAM 算法性能

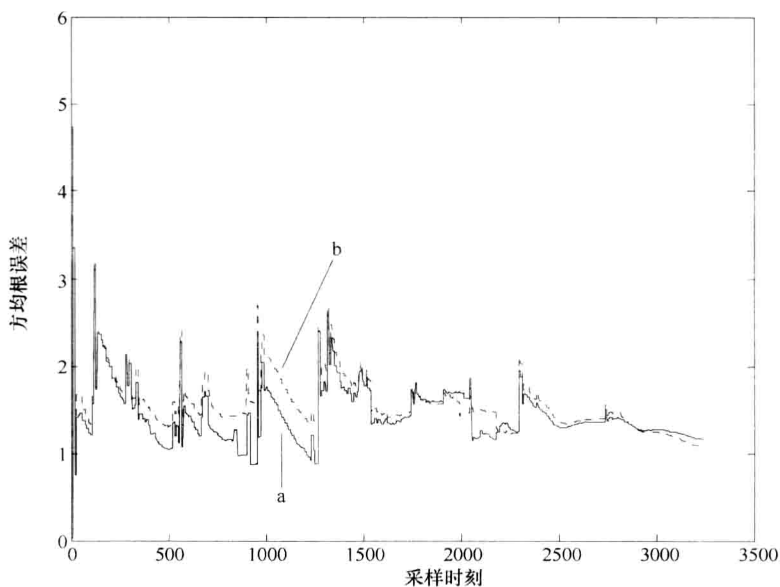


图 7.15　模糊监督器分别由差分进化算法(a)和 PSO 算法训练(b)的模糊监督的基于 EKF 的 SLAM 算法性能评估比较

接下来，对由差分进化算法调节的模糊监督器和由 PSO 算法调节的模糊监督器进行性能比较。在此是针对传感器统计为 $\sigma_r = 0.01\text{m}$，$\sigma_b = 15.0°$时的示例进行性能比较，其中 PSO 算法中采用了如式(7.35)所示的线性递减惯性权重。为使得基于差分进化算法和基于 PSO 算法参数调节的算法进行统一比较，在此考虑了以下因素：每个算法的候选解或粒子数相同(如 20)，每次优化算法运行的最大迭代次数相同(在此取 10)；每个优化算法中相应的初始种群维数的初始范围相同。通常情况下，初始惯性权重 W_{initial}为 0.9，惯性权重的范围为 2.5e－4 的 PSO 算法对于基准优化函数较为理想。在本示例中，采用了 $W_{\text{initial}} = 0.9$ 以及惯性权重缓慢减小和急剧减小的 PSO。图 7.14 给出了当基于 PSO 的训练过程中惯性权重斜率分别为 2.0e－4、2.5e－4、5.0e－4、4e－2 和 5e－2 时 PSO 算法的相应性能，这是通过机器人增量运动的每个采样时刻估计$\hat{\boldsymbol{x}}$的方均根误差来反映的。由此可看出，最佳性能是在值为 2.5e－4 时获得的。图 7.15 给出了本例条件下，基于自适应 EKF 的 SLAM 算法中最佳 PSO 调节和最佳 DE 调节模糊监督器的估计性能比较。可看到 DE 调节算法的性能，在大多数采样时刻，具有较小的方均根估计误差。该过程表明了对于基于 EKF 的 SLAM 问题，采用 DE 调节的模糊监督器具有很大作用。然而还应注意到，该性能会根据所选环境和传感器统计值的变化而变化。

7.6 本章小结

本章讨论了 SLAM 在移动机器人导航中的重要性，并首先详细介绍了基于 EKF 的 SLAM 算法。接下来，认为在传感器统计值的先验知识有误时，系统性能会下降，并表明了为何利用模糊/模糊神经网络辅助或监督能够显著提高算法性能。通常，在关联统计模型完全已知时，认为 EKF 对 SLAM 算法而言是一种不错的选择。然而当统计信息不够准确时，性能就可能变得根本无法预测或下降。基于模糊/模糊神经网络监督器的系统能够以错误的统计值开始，在线调整矩阵 $\boldsymbol{R}$，使得新息序列理论值与实际值之间的误差最小。采用一个基于进化优化算法的训练过程，可自动学习模糊神经网络系统的自由参数。本章表明两种常用的进化优化方法(即 PSO 和 DE)如何能够成功地用于实现该目标。针对几种具有不同传感器统计值错误信息的标准环境进行了性能评估。在许多情况下，传统的基于 EKF 的 SLAM 算法表现出性能不可靠并急剧下降，而结合模糊/模糊神经网络可显著提高 EKF 的性能，并在各种示例的每种情况下都能具有鲁棒且准确的性能。

致谢 该研究工作得到了日本国外研究人员博士后研究基金 JSPS 的部分资

助。另外还得到了印度技术教育委员会 RPS 项目的部分资助(资助号：8023/BOR/RPS-89/2006-07)。

参考文献

[1] Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Tran. Robotics and Automation 17(3), 229–241 (2001)

[2] Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico (2003)

[3] Grisetti, G., Stachniss, C., Burgard, W.: Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, pp. 2443–2448 (2005)

[4] Smith, R., Cheeseman, P.: On the representation and estimation of spatial uncertainty. International Journal of Robotics Research 5(4) (1986)

[5] Moutarlier, P., Chatila, R.: Stochastic multisensory data fusion for mobile robot location and environment modeling. In: 5th Int. Symposium on Robotics Research, Tokyo (1989)

[6] Davison, A.J.: Mobile Robot Navigation Using Active Vision. PhD Thesis, Univ. of Oxford (1998)

[7] Bailey, T.: Mobile Robot Localization and Mapping in Extensive Outdoor Environments. PhD Thesis, Univ. of Sydney (2002)

[8] Davison, A.J., Murray, D.W.: Simultaneous localization and map-building using active vision. IEEE Tran. Pattern Analysis and Machine Intelligence 24(7), 865–880 (2002)

[9] Guivant, J., Nebot, E.: Optimization of the simultaneous localization and map-building algorithm and real-time implementation. IEEE Tran. Robotics and Automation 17(3), 242–257 (2001)

[10] Guivant, J., Nebot, E.: Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms. IEEE Tran. Robotics and Automation 19(4), 749–755 (2003)

[11] Williams, S.B., Newman, P., Dissanayake, G., Durrant-Whyte, H.: Autonomous underwater simultaneous localization and map building. In: Proc. IEEE International Conference on Robotics and Automation, San Francisco, CA, vol. 2, pp. 1792–1798 (2000)

[12] Chong, K.S., Kleeman, L.: Feature-based mapping in real, large scale environments using an ultrasonic array. International Journal of Robotic Research 18(2), 3–19 (1999)

[13] Bosse, M., Leonard, J., Teller, S.: Large-scale CML using a network of multiple local maps. In: Leonard, J., Tardós, J.D., Thrun, S., Choset, H. (eds.) Workshop Notes of the ICRA Workshopon Concurrent Mapping and Localization for Autonomous Mobile Robots (W4), Washington, DC. ICRA Conference (2002)

[14] Thrun, S., Fox, D., Burgard, W.: A probabilistic approach to concurrent mapping and localization for mobile robots. Machine Learning 31, 29–53 (1998); also appeared in Autonomous Robots 5, 253–271 (joint issue)

[15] Williams, S., Dissanayake, G., Durrant-Whyte, H.F.: Towards terrain-aided navigation for underwater robotics. Advanced Robotics 15(5) (2001)

[16] Thrun, S., Hähnel, D., Ferguson, D., Montemerlo, M., Triebel, R., Burgard, W., Baker, C., Omohundro, Z., Thayer, S., Whittaker, W.: A system for volumetric robotic mapping of abandoned mines. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA (2003)

[17] Castellanos, J.A., Montiel, J.M.M., Neira, J., Tardós, J.D.: The SPmap: A probabilistic framework for simultaneous localization and map building. IEEE Transactions on Robotics and Automation 15(5), 948–953 (1999)

[18] Paskin, M.A.: Thin junction tree filters for simultaneous localization and mapping. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico (2003)

[19] Thrun, S., Koller, D., Ghahramani, Z., Durrant-Whyte, H., Ng, A.Y.: Simultaneous mapping and localization with sparse extended information filters. In: Boissonnat, J.-D., Burdick, J., Goldberg, K., Hutchinson, S. (eds.) Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics, Nice, France (2002)

[20] Neira, J., Tardós, J.D.: Data association in stochastic mapping using the joint compatibility test. IEEE Transactions on Robotics and Automation 17(6), 890–897 (2001)

[21] Shatkay, H., Kaelbling, L.: Learning topological maps with weak local odometric information. In: Proceedings of IJCAI 1997. IJCAI, Inc. (1997)

[22] Araneda, A.: Statistical inference in mapping and localization for a mobile robot. In: Bernardo, J.M., Bayarri, M.J., Berger, J.O., Dawid, A.P., Heckerman, D., Smith, A.F.M., West, M. (eds.) Bayesian Statistics 7. Oxford University Press, Oxford (2003)

[23] Montemerlo, M., Thrun, S.: Simultaneous localization and mapping with unknown data association using Fast SLAM. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan (2003)

[24] Hu, W., Downs, T., Wyeth, G., Milford, M., Prasser, D.: A modified particle filter for simultaneous robot localization and Landmark tracking in an indoor environment. In: Proc. Australian Conference on Robotics and Automation (ACRA), Canberra, Australia (2004)

[25] Frese, U., Larsson, P., Duckett, T.: A multilevel relaxation algorithm for simultaneous localization and mapping. IEEE Tran. Robotics 21(2), 196–207 (2005)

[26] Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada, AAAI (2002)

[27] Lu, F., Milios, E.: Globally consistent range scan alignment for environment mapping. Autonomous Robots 4, 333–349 (1997)

[28] Mehra, R.K.: On the identification of variances and adaptive Kalman filtering. IEEE Tran. Automatic Control AC-15(2), 175–184 (1970)

[29] Fitzgerald, R.J.: Divergence of the Kalman filter. IEEE Tran. Automatic Control AC-16(6), 736–747 (1971)

[30] Sinha, N.K., Tom, A.: Adaptive state estimation for systems with unknown noise covariances. International Journal of Systems Science 8(4), 377–384 (1977)

[31] Bellanger, P.R.: Estimation of noise covariance matrices for a linear time-varying stochastic process. Automatica 10, 267–275 (1974)

[32] Dee, D.P., Cohn, S.E., Dalcher, A., Ghil, M.: An efficient algorithm for estimating noise covariances in distributed systems. IEEE Tran. Automatic Control AC-30(11), 1057–1065 (1985)

[33] Reynolds, R.G.: Robust estimation of covariance matrices. IEEE Tran. Automatic Control 32(9), 1047–1051 (1990)

[34] Morikawa, H., Fujisaki, H.: System identification of the speech production process based on a state-space representation. IEEE Trans. Acoust., Speech, Signal Processing ASSP-32, 252–262 (1984)

[35] Noriega, G., Pasupathy, S.: Adaptive estimation of noise covariance matrices in real-time preprocessing of geophysical data. IEEE Trans. Geoscience and Remote Sensing 35(5), 1146–1159 (1997)

[36] Kobayashi, K., Cheok, K.C., Watanabe, K., Munekata, F.: Accurate differential global positioning system via fuzzy logic Kalman filter sensor fusion technique. IEEE Tran. Industrial Electronics 45(3), 510–518 (1998)

[37] Loebis, D., Sutton, R., Chudley, J., Naeem, W.: Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system. Control Engineering Practice 12, 1531–1539 (2004)

[38] Wu, Z.Q., Harris, C.J.: An adaptive neurofuzzy Kalman filter. In: Proc. 5th International Conference on Fuzzy Sets and Systems FUZZ-IEEE 1996, vol. 2, pp. 1344–1350 (September 1996)

[39] Sasiadek, J.Z., Wang, Q., Zeremba, M.B.: Fuzzy adaptive Kalman filtering for INS/GPS data fusion. In: Proc. 15th International Symposium on Intelligent Control (ISIC 2000), Rio, Patras, Greece (July 2000)

[40] Clerc, M., Kennedy, J.: The particle swarm-explosion, stability and convergence in a multidimensional complex space. IEEE Tran. Evolutionary Computation 6(1), 58–73 (2002)

[41] Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congr. Evolutionary Computation, pp. 1945–1950. IEEE Service Center, Piscataway (1999)

[42] `http://www.acfr.usyd.edu.au/homepages/academic/tbailey/software/software.html`

[43] Brown, R.G., Hwang, P.Y.C.: Introduction to Random Signals and Applied Kalman Filtering, 3rd edn. John Wiley and Sons, USA (1997)

[44] Chatterjee, A., Matsuno, F.: A neuro-fuzzy assisted extended Kalman filter-based approach for Simultaneous Localization and Mapping (SLAM) problems. IEEE Transactions on Fuzzy Systems 15(5), 984–997 (2007)

[45] Chatterjee, A.: Differential evolution tuned fuzzy supervisor adapted extended kalman filtering for SLAM problems in mobile robots. Robotica 27(3), 411–423 (2009)

[46] Storn, R.: On the usage of differential evolution for function optimization (1996)

[47] Storn, R., Price, K.: Minimizing the real functions of the ICEC 1996 contest by differential evolution (1996)

[48] `http://www.icsi.berkeley.edu/~storn/code.html` (last accessed June 24, 2008)

第 8 章　基于视觉的移动机器人 SLAM㊀

摘要：本章是对第 7 章内容的扩展和补充，讨论了如何在室内环境下真正实现第 7 章所介绍的移动机器人 SLAM。所开发的机器人系统具有一个包含两个摄像头的视觉系统，可用于图像特征识别与跟踪。

8.1　简介

如第 7 章所述，基于 EKF 的方法已被普遍认为是解决移动机器人 SLAM 问题的最适合的方法[1-7]。EKF 在解决 SLAM 问题上最基本的能力在于迭代评估的方法，并由此建立一个通过某些导航点直接对机器人导航的环境增广地图。在此假设机器人位姿的初始位置和所要构建的地图均未知，并将地图看作一种增广估计状态来逐步构建地图，这时地图中只包含一些在环境中采集的特征或路标的位置，以及机器人的位姿。这些状态的估计都与其中的某些不确定性相关联，并以一种误差协方差矩阵的形式保存。在第 7 章已详细讨论了基于 EKF 的 SLAM 算法。本章将讨论如何基于视觉感知来实现移动机器人的 SLAM。

另外，还要注意到在实际环境中真正实现 SLAM 算法来构建一个有意义的地图是一项艰巨的任务。该系统的准确性很大程度上取决于所用的传感器。正如我们所知，里程计会受车轮打滑的影响，声纳传感器的分辨率较低而不是一个非常精确的系统，并受环境干扰的影响，红外传感器只能用于短距离的测量，而激光测距仪成本较高，而且由于更新速率较低而导致响应较慢，GPS 的性能会由于卫星信号的遮挡而受到影响，造成准确性和更新率降低。因此近年来，摄像头和电脑作为一种有吸引力的、可行的实时方案来构建这种机器人定位系统[3,5]。同时，成本相对低廉，并能够对机器人平台的导航环境提供灵活解释。然而直到目前，有关基于视觉感知的 SLAM 算法的研究还很少，主要原因是开发该系统以及在实际中达到较高的准确性是一项艰巨的任务。

㊀ 本章基于 Avishek Chatterjee、Olive Ray、Amitava Chatterjee 和 Anjan Rakshit 于 2011 年 7 月在 Expert Systems with Applications 期刊上第 38 卷第 7 期 8266 ~ 8274 页上发表的“Development of a Real - Life EKF based SLAM System for Mobile Robots employing Vision Sensing”论文。

本章将对用于室内环境中地图构建的 SLAM 算法详细介绍其具体实现过程[15]，在此是针对一种已在之前章节中介绍过的称为 KOALA 的差动驱动移动机器人。本书所提方法的一个重要特征是在这种独立系统中采用了基于计算机视觉的感知系统来构建地图。该视觉系统采用了两个摄像头在所采集的图像中进行特征识别，以及在随后的图像帧中进行特征跟踪。这种对于场景标识和障碍识别必不可少的系统有助于开发更适合的导航算法，进行避障和/或构建机器人导航的环境地图。特征跟踪方法是基于上一窗口与当前窗口的亮度差二次方和最小来决定当前窗口是否是滑动后过去窗口。另外，该系统还具有在机器人坐标中计算路标 3D 距离的模块，这样就可确定在地图中的机器人位置，并在地图中保存机器人当前位置和路标坐标。该系统已在本实验室中真正用于构建导航地图，同时在室内环境中所构建的地图具有较高的精度。

8.2　差动驱动 KOALA 机器人的动态状态模型

在 7.2 节中已详细介绍了基于 EKF 的 SLAM 算法。在此，为将该理论应用于实际的 KOALA 机器人，首先本节将建立差动驱动的 KOALA 机器人的动态模型。当然这也可从逻辑上扩展到其他类似的移动机器人。这时，主要有两个独立变量决定了小车的运动，即左轮的电动机转速和右轮的电动机转速。然而在此分别将机器人几何中心的直线位移量和绕中心垂直轴的旋转量这两个衍生变量看作主导变量。进行域转换的原因是由于在旋转过程中会严重变形，因此如果将旋转看作一个变量，将会引入误差。在线性、平移运动中不会存在这种问题，只是误差源或不确定性不同，例如，里程计编码器标定不准确、车轮旋转时打滑等。在此，假设机器人不会同时进行旋转运动和平移运动。

在建模时，应注意到机器人总是沿圆弧运动的。对于线性平移运动，曲率为零，而对于旋转运动，曲率半径为零。图 8.1 给出了机器人的运动机制，此时

$$\overset{\frown}{A}\overset{\frown}{B}\overset{\frown}{C} = s \tag{8.1}$$

$$\theta = \frac{s}{r} = K_1(\text{右轮旋转量} - \text{左轮旋转量}) \tag{8.2}$$

$$s = (K_2/2)(\text{右轮旋转量} + \text{左轮旋转量}) \tag{8.3}$$

由式(8.2)和式(8.3)可得 s，同时，根据车轮编码器读数可直接获得 θ。由此可得，$r=\frac{s}{\theta}$和$\overline{AC}=2r\sin\frac{\theta}{2}$。然后，当已知机器人的初始位置 ϕ 时，可将 AC 分解为 x 分量和 y 分量。因此可得

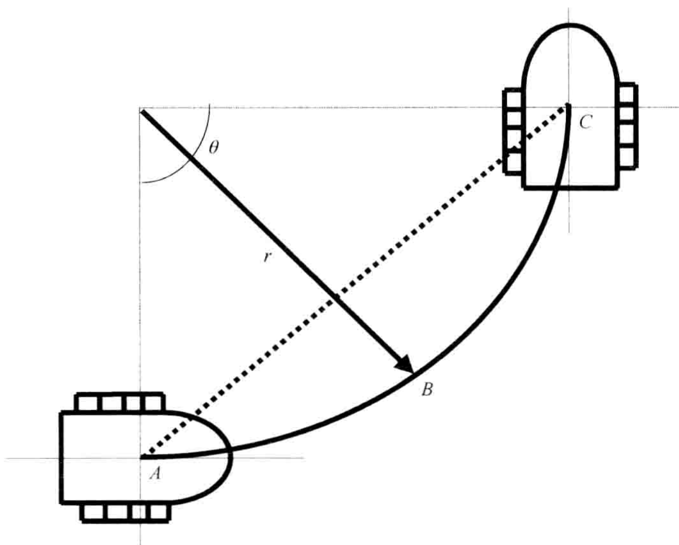


图 8.1　KOALA 机器人的运动机制

$$\left.\begin{aligned} dx &= 2r\sin\frac{\theta}{2}\cos\phi \\ dy &= 2r\sin\frac{\theta}{2}\sin\phi \\ d\phi &= \theta \end{aligned}\right\} \tag{8.4}$$

在 $\theta\to 0°$ 的情况下建模时，由于 $r=\frac{s}{\theta}\to\infty$，因此会产生一个逻辑问题。在此，当 $\theta<5°$ 时，假设 $\overline{AC}=s$。对于 KOALA 机器人的直线位移量 D 和旋转量 θ，可由下式建立机器人的最终动态模型：

$$\left.\begin{aligned} \Delta x &= D\cos\phi \\ \Delta y &= D\sin\phi \\ d\phi &= \theta \end{aligned}\right\} \Rightarrow \left.\begin{aligned} x_{k+1} &= x_k + D\cos\phi \\ y_{k+1} &= y_k + D\sin\phi \\ \phi_{k+1} &= \phi_k + \theta \end{aligned}\right\} \tag{8.5}$$

由此，可计算雅各比矩阵和协方差矩阵：

$$\nabla \boldsymbol{f}_u = \begin{bmatrix} \frac{\partial \Delta x}{\partial D} & \frac{\partial \Delta x}{\partial \theta} \\ \frac{\partial \Delta y}{\partial D} & \frac{\partial \Delta y}{\partial \theta} \\ \frac{\partial \Delta \phi}{\partial D} & \frac{\partial \Delta \phi}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \tag{8.6}$$

$$\nabla f_{x_v} = \begin{bmatrix} 1 & 0 & -D\sin\phi \\ 0 & 1 & D\cos\phi \\ 0 & 0 & 1 \end{bmatrix} \tag{8.7}$$

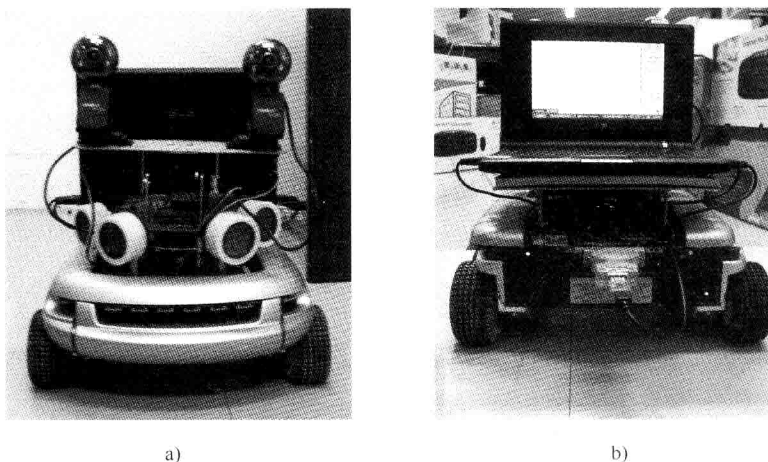
$$Q = \begin{bmatrix} \sigma D^2 & 0 \\ 0 & \sigma\theta^2 \end{bmatrix}$$

式中　$\sigma D = D \times$ 单位位移的标准偏差；

$\sigma\theta = \theta \times$ 单位旋转的标准偏差。

8.3　基于视觉感知的图像特征识别、特征跟踪和每个特征的 3D 距离计算

在本书所提出的 SLAM 算法中，利用视觉感知进行“观测”。所购买的基本配置的 KOALA 机器人中已具有一些内置传感器，如车轮的增量编码器和红外传感器，后来还集成了一些如声纳、无线通信模块、传感器云台扫描系统、视觉系统、控制四个自由度的伺服电动机、计算平台等附件。所有的集成工作都是在实验室内完成的。图 8.2 给出了专门用于进行视觉感知 SLAM 的集成模式下的 KOALA 机器人。



a)　　　　　　b)

图 8.2　KOALA 移动机器人

a）配置有一些内置传感器的原装机器人　b）在实验室集成了一些附件的改装机器人

在具体实现中，视觉感知采用了两个网络摄像头，如图 8.2a 所示，其主要目的是用于在一个双摄像头的视觉系统中实现图像特征选择，并跟踪所选择的特征和计算所选择特征的 3D 距离[16]。根据从每个摄像头获得的实时视频采集的图像帧中选择合适的图像块或窗口来进行特征识别，这能够保证在后续图像

帧中具有较高的跟踪能力。在实际应用中，具有丰富边缘信息的图像块更适用于特征跟踪，因此这些图像块（在本系统中是静态的）可作为用于地图构建的最佳路标。不同视角下（对于本系统，即左右两个摄像头分别采集的左图和右图）的特征匹配计算是获得深度信息所必需的前提条件。系统首先在左摄像头采集的图像中执行一个特征识别算法来确定合适的矩形图像块或最适合作为特征跟踪的窗口（具有足够纹理的图像块），然后在右摄像头采集的图像中进行特征跟踪。该图像跟踪系统的开发思想来自于 KLT 跟踪器[10,13]。该跟踪器总是优先选择跟踪一个像素窗口或图像块而不是单个像素点，这是因为几乎不可能跟踪单个像素点，除非与所有相邻像素点之间的亮度值完全不同。同时，由于像素亮度值会因存在噪声而变化，因此跟踪结果也会非常混乱。由此，在亮度曲线分布的基础上，通过在一幅图像帧中特征之间的距离最小来选择 N 个特征窗口。对于一个二维图像函数 $f(x, y)$，其梯度是一个矢量，且每个窗口 $\boldsymbol{G}$ 的梯度可沿 x 和 y 方向进行计算：

$$\boldsymbol{G} = \begin{bmatrix} g_{xx} & g_{xy} \\ g_{xy} & g_{yy} \end{bmatrix} = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \tag{8.8}$$

根据计算矩阵 $\boldsymbol{G}$ 的特征根 λ_1 和 λ_2 来评估所选窗口是否适合作为一个特征窗口，若

$$\min(\lambda_1, \lambda_2) > \lambda \tag{8.9}$$

则可作为特征窗口。其中，λ 是一个预设阈值[14]。两个较小的特征根意味着窗口中的亮度曲线分布大致相同，而一个较大和一个较小的特征根则对应着一个单向纹理模板。另一方面，两个较大的特征根表示是角点或椒盐纹理[11,16]。

一旦特征选择完成，接下来就要在图像序列中一帧帧地跟踪这些特征[11-13]。类似于参考文献［11］，在此计算使得两帧图像中窗口间图像亮度差的二次方和最小的一个特征窗口的中心位移 $\boldsymbol{dp} = [dxp, dyp]^{\mathrm{T}}$。帧间运动较小时，两帧图像中特征的运动可由一个纯位移模型近似。然而帧间运动较大时，由线性变形和纯位移构成的一个仿射模型被认为更准确。在此，根据其中包括一个表示图像帧中线性变形的仿射运动模型以及特征平移的形变矩阵的差异度测量，跟踪过程中特征跟踪的质量更高。图像中像素点的运动可由下式表示：

$$J(\boldsymbol{Axp} + \boldsymbol{dp}) = I(\boldsymbol{xp}) \tag{8.10}$$

式中 J——当前图像；

I——原始图像；

$\boldsymbol{A} = \boldsymbol{I} + \boldsymbol{D}$（$\boldsymbol{I}$ 为一个 2×2 的单位矩阵，$\boldsymbol{D}$ 为形变矩阵）；

$\boldsymbol{dp}$——平移矢量。

由此，可利用 $w(\boldsymbol{xp})$，即一个参考文献［11］中的权值函数（通常选取一个单位函数或高斯函数来强调窗口中心区域）来计算差异度：

$$\varepsilon = \iint_W [J(\boldsymbol{Axp} + \boldsymbol{dp}) - I(\boldsymbol{xp})]^2 w(\boldsymbol{xp}) \mathrm{d}\boldsymbol{xp} \tag{8.11}$$

利用两个窗口图像亮度值之间的牛顿—拉夫逊（Newton-Raphson）最小化算法来迭代搜索一帧新的图像中新的特征窗口中心点位置，接下来需对系统进行求解来获得 $\boldsymbol{dp}$：

$$\boldsymbol{Gdp} = \boldsymbol{e} \tag{8.12}$$

式中 $\boldsymbol{G} = \int(\boldsymbol{gg}^{\mathrm{T}} w)\mathrm{d}a, \boldsymbol{G}$ = 二阶加权系数矩阵(2 × 2)；

$\boldsymbol{e}$ = 加权亮度值误差矢量（2 × 1）$\left(\boldsymbol{e} = \left(\int_W (I - J)\right)\boldsymbol{g}w\mathrm{d}a\right)$；

$\boldsymbol{dp}$ = 平移矢量（2 × 1）（$\boldsymbol{dp} = [dxp \quad dyp]^{\mathrm{T}}$）；

$\boldsymbol{g}$ = 梯度矢量（2 × 1）$\left(\boldsymbol{g} = \left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y}\right]^{\mathrm{T}}\right)$。

该迭代算法通过在每次迭代中对$\begin{bmatrix} g_{xx} & g_{xy} \\ g_{xy} & g_{yy} \end{bmatrix}\begin{bmatrix} dxp \\ dyp \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \end{bmatrix}$进行求解，并计算图像中新的窗口中心来求解式（8.12），在本次迭代中对 $x_{\text{p_tracked}} = x_{\text{p_tracked}} + dxp$、$y_{\text{p_tracked}} = y_{\text{p_tracked}} + dyp$ 进行跟踪。

根据有关所用摄像头和云台几何形状的数据，可得到所跟踪路标的 3D 距离[3,9,14]。为获得立体视觉的深度信息，需要两个摄像头的视线在场景中的 P 点相交，然后根据该信息，可获得世界坐标系（WCS）中所观测的场景点的三维坐标信息。本系统中的距离计算模块基于 Andrew J. Davison 所提出的小孔成像模型[3]，即利用了常用的摄像头标定矩阵和透视投影方程，并结合“最近中点法”。图 8.3 给出了本实验室所设计和开发的主动云台的正视图，其中 $\boldsymbol{H}$ = 云台中心与地面的垂直距离，$\boldsymbol{I}$ = 左摄像头聚焦轴和右摄像头聚焦轴之间的水平距离，c = 俯仰轴与摄像头光轴交点之间每个聚焦轴的偏移量。

一旦在左、右摄像头图像中识别并跟踪到新路标或图像块，就可利用基于 EKF 的 SLAM 算法中新路标的初始化步骤在地图中进行初始化。类似地，左、右摄像头图像中图像块的识别与跟踪（即在之前位置上机器人采集的图像中所识别的图像块）将构成基于 EKF 的 SLAM 算法中的重新观察步骤。在该步骤中，根据卡尔曼滤波的“预测”步骤来计算路标点的估计位置，然后通过执行卡尔曼滤波算法中的“观察和更新”步骤来进一步修正。

基于视觉感知的 EKF-SLAM 算法的具体实现步骤如算法 8.1 所示。在此，要求机器人经过某些导航点，并直接构建其周围的环境地图。为实现该功能，机器人应先运动一定距离，并采集几帧图像来同时进行路标观测和自定位。为构建机器人前方、左侧和右侧的环境地图，分别在水平角 0°、$+\theta°$和 $-\theta°$位置

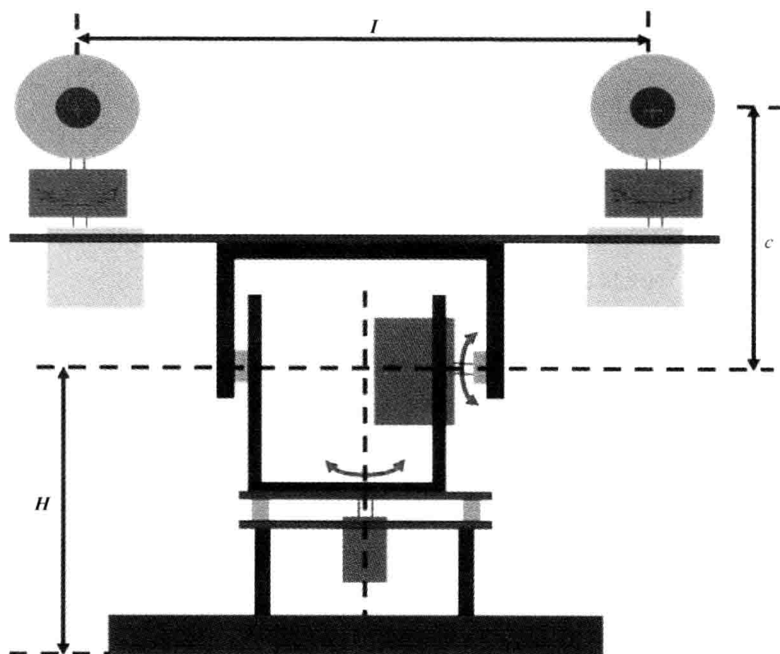


图 8.3 本实验室设计的包括传感器扫描云台系统、两个网络摄像头和四个用于四自由度独立控制（水平控制、俯仰控制、左摄像头聚焦控制和右摄像头聚焦控制）的伺服电动机的主动云台的前视图

处采集图像。由此，在 EKF 算法的观测步骤中，机器人在其前方、左侧和右侧的环境中识别并获取特征/路标。在机器人的前向运动过程中，通过视觉传感器在几个水平角方向上获取和跟踪路标来执行“预测”步骤，并以迭代形式执行 EKF 算法的“校正和更新”步骤，直到到达最后一个导航点。在最后一次迭代中所构建的地图看作可应用于在同一环境中执行其他任务时机器人所构建的最终地图。

算法 8.1 在室内环境下（本实验室），利用视觉传感器在 KOALA 机器人上实现的基于 EKF 的实际 SLAM 算法

步骤 1 指定机器人导航所需的导航点和机器人初始位姿。

步骤 2 机器人运动一段距离，并执行 EKF 算法的预测步骤。

步骤 3 在水平角 0°处从左、右两个摄像头的连续视频序列中采集图像，并进行特性识别、特征跟踪和计算特征点与机器人之间的距离。

步骤 4 在水平角 $+\theta°$处，重复执行步骤 3。

步骤 5 在水平角 $-\theta°$处，重复执行步骤 3。

步骤 6　对于步骤 3 ~ 步骤 5 所观测到的新特征/路标，在地图中进行初始化。

步骤 7　对于步骤 3 ~ 步骤 5 观测到的新特征/路标，执行 EKF 算法的观察与更新步骤，并修正已构建的地图。

步骤 8　执行步骤 2 ~ 步骤 7，直到机器人到达最后一个导航点。

步骤 9　保存机器人所构建的最终地图，并作为环境地图。

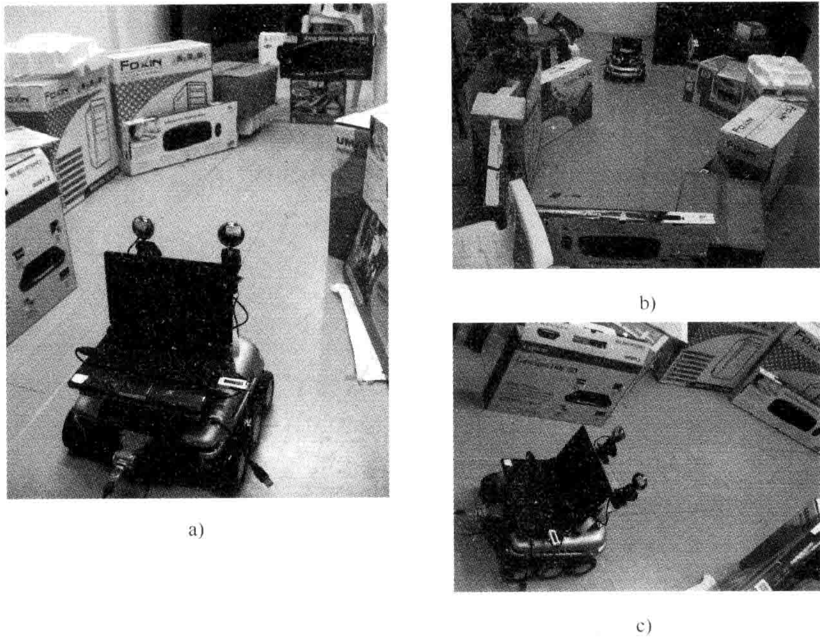
8.4　实际性能评估

如上所述，KOALA 机器人的大小为 32cm × 32cm，是由瑞士 K-team 公司研制的六轮差动驱动小车。在前面的章节中已介绍了 KOALA 机器人中的硬件控制是由车载微控制器（16MHz Motorola 68331@22MHz）实现的[8]。为在机器人系统中增加水平、俯仰、左摄像头聚焦和右摄像头聚焦四个自由度，开发了一个基于 PIC 16F876A 微控制器的系统，该系统工作在中断驱动模式，并与 Motorola 控制器构成一个主/从配置。这种用于与实际机器人外设接口的 PIC 微控制器系统能够大大提高实际应用的灵活性，具体开发过程已在第 2 章中详细介绍。

图 8.4 给出了一个要求机器人通过几个指定导航点进行导航并通过执行基于视觉的 SLAM 算法来构建地图的室内环境。为评价系统性能，在此绘制了一个包含 100 个方块，其中每个方块大小为 20cm × 20cm 的栅格，即 2m × 2m 的导航区域。

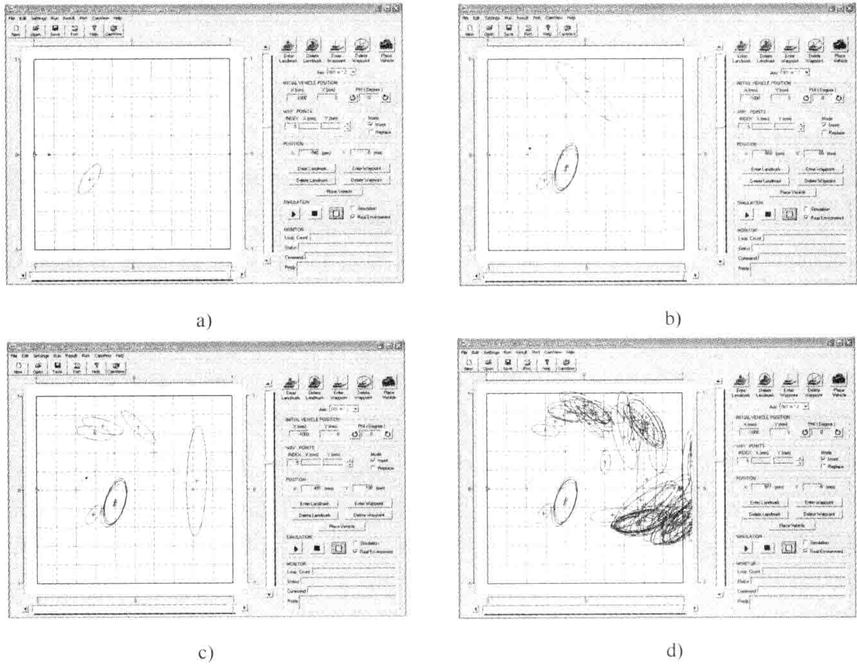
图 8.5 给出了实验室环境下实际执行 EKF-SLAM 算法时的 GUI 软件界面。图 8.5 的不同图像中给出了在利用 EKF-SLAM 算法构建增广地图的迭代过程中所识别的路标，并集成在所构建的地图中。点线表示机器人导航中增加路标点的理想路径。“淡蓝色三角”表示机器人的初始位置，如图 8.4 所示，在实现过程中该初始位置为 $(z, x, \phi)^{\mathrm{T}} = (-100, 0, 0)^{\mathrm{T}}$。对于此处的具体实现，符号 z、x 和 ϕ 与参考文献 [3] 中所用的符号一致。因此，z 方向和 x 方向分别对应 x 方向和 y 方向，正如理论中所指定的一样。在导航过程中，机器人识别其周围环境中的路标，并初始化其位置或在地图中修正其位置。随着机器人不断前进，识别的路标个数以及地图大小也不断增加。地图中的“十字”表示所识别的路标的 2D 位置。图 8.5d 给出了 KOALA 机器人测试运行后得到的最终地图。

图 8.6 给出了实际中采集图像的 GUI 软件开发界面，包括 KOALA 机器人的位置表示、特征提取和跟踪算法的性能演示以及路标识别等。实线方块中的图像块看作新路标，虚线方块中的图像块看作重新观察的路标。界面中还显示了每个跟踪路标与机器人的 3D 计算距离。



a) b) c)

图 8.4 机器人导航并执行 EKF SLAM 算法的环境



a) b) c) d)

图 8.5 EKF-SLAM 算法的不同步骤中用于构建地图的真实路标识别

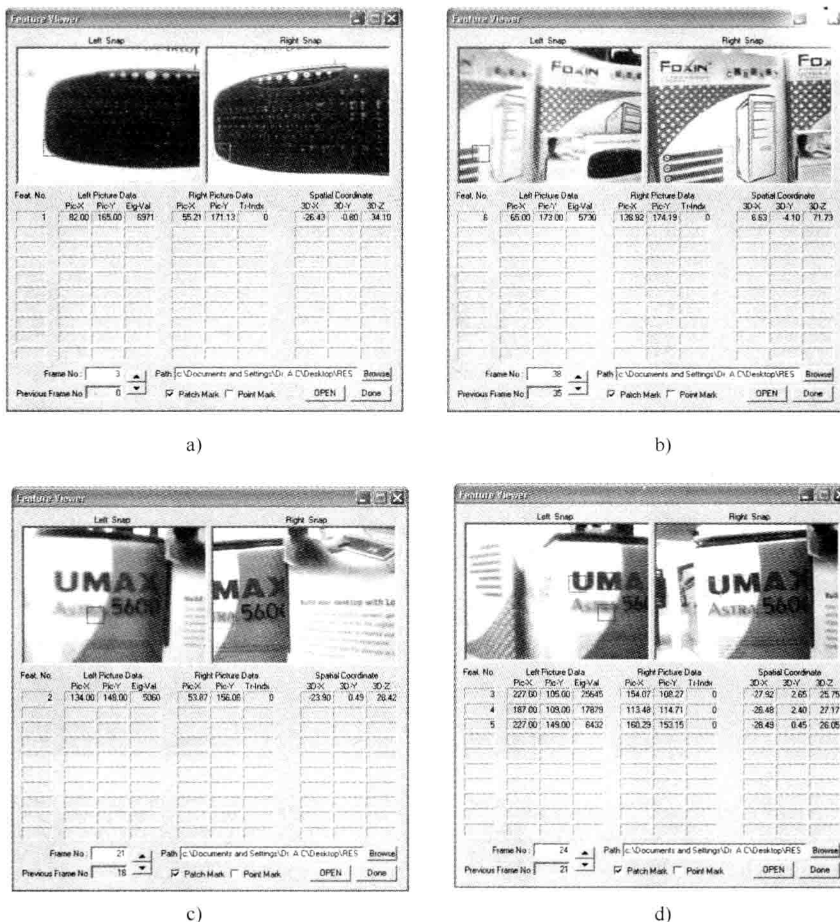


图 8.6　KOALA 机器人在环境中测试运行过程中，在一些有代表性的位置上进行特征提取、特征跟踪以及所跟踪特征与机器人之间 3D 距离计算的实验结果

图 8.7 给出了实际环境中特征/路标识别与跟踪的三种示例。网格中和垂直方向上的线以及点有助于标识和获得环境中的实际位置。空心圆圈表示相对于环境中识别的图像块的实际对象。所构建的地图中这些路标的估计位置如图 8.5 所示，表明在实际 2D 位置与地图中大多数估计 2D 位置之间具有较小的偏差，然而这在实际实验中是可以理解的。表 8.1 给出了三种情况下这些路标的实际位置和估计位置，如图 8.7 所示。

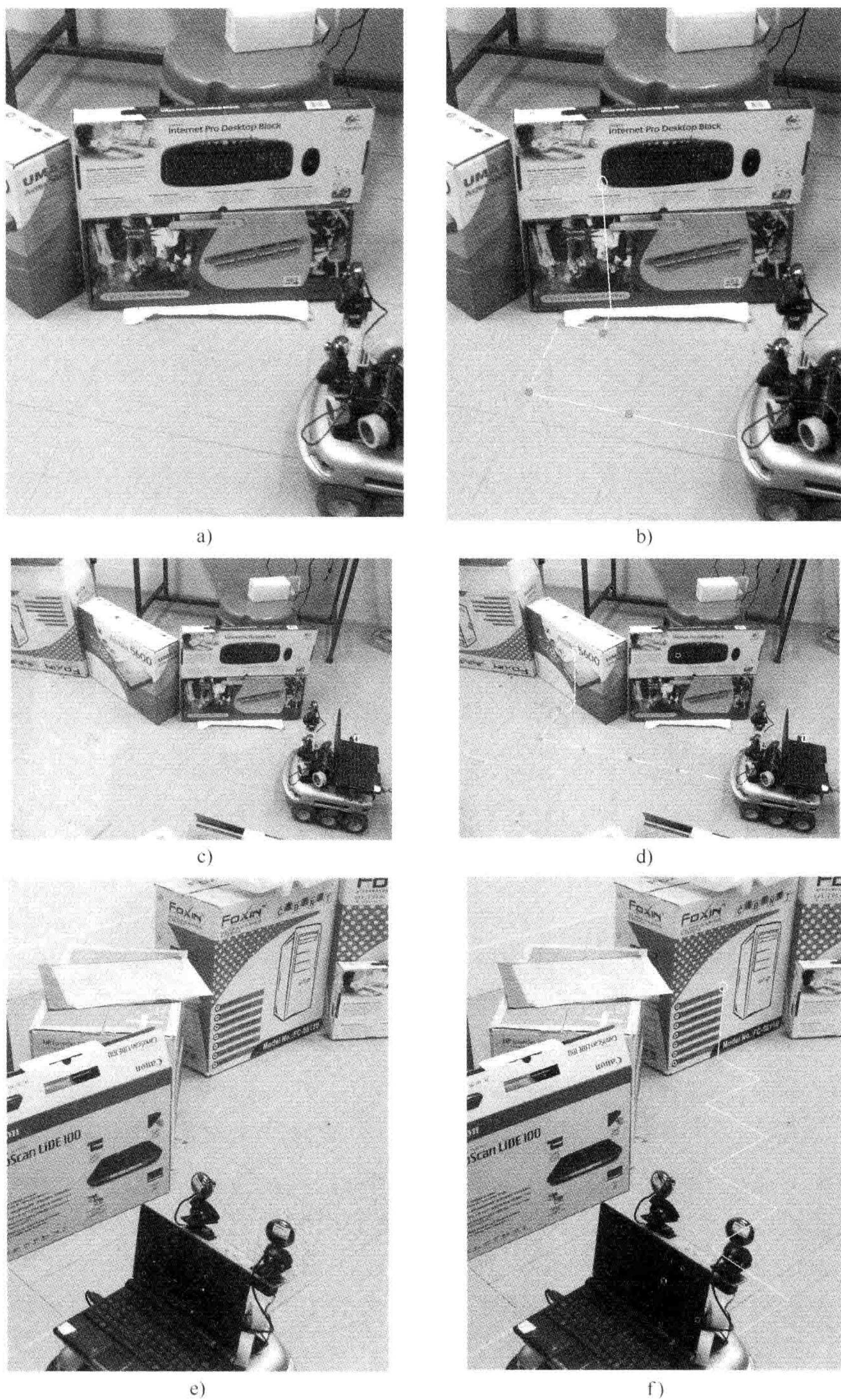


图 8.7 真实环境中路标识别和跟踪的三种示例情况

表 8.1　对于图 8.7 所示的三个实际路标，比较基于视觉感知的 EKF-SLAM 算法的性能

SI 编号	路标描述	估计位置/cm		实际位置/cm	
		z 坐标	x 坐标	z 坐标	x 坐标
1	图 8.6[①] a 和图 8.6[①] b 中的路标（键盘的左下角）	-43	-26	-47	-27
2	图 8.6[①] c 和图 8.6[①] d 中的路标（UMAX 盒子中字母“A”的左下角）	-18	-18	-10	-23
3	图 8.6[①] e 和图 8.6[①] f 中的路标（FOXIN 盒子中粗条线的右上角）	4	70	2	74

① 此处原书有误，应为图 8.7。——译者注

8.5　本章小结

本章介绍了室内环境下利用基于两个网络摄像头的视觉感知机制的移动机器人 SLAM 的理论与具体实现。该系统体现了一个算法的成功实现，首先从安装在自主开发的 KOALA 移动机器人主动云台上的两个摄像头所获得的实时视频采集图像帧中进行图像特征识别；然后在后续图像帧中跟踪当前图像帧中所识别的特征/路标，并集成在所构建的地图中；最后在从机器人坐标系中计算的路标距离基础上，利用 3D 距离计算模块来计算路标在世界坐标系中的实际坐标。该系统成功地在实验室环境下进行了测试运行，实验表明地图中的路标估计位置与实际路标位置之间的偏差很小。希望这种成功实现能够启发更多的读者在更复杂环境下和室外环境下实现类似的地图构建系统。

致谢　本章所介绍的研究工作得到了印度技术教育委员会 RPS 项目的资助（资助号：8023/BOR/RPS-89/2006-07）。

参考文献

[1] Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Tran. Robotics and Automation 17(3), 229–241 (2001)
[2] Smith, R., Cheeseman, P.: On the representation and estimation of spatial uncertainty. International Journal of Robotics Research 5(4) (1986)
[3] Davison, A.J.: Mobile Robot Navigation Using Active Vision. PhD Thesis, Univ. of Oxford (1998)
[4] Bailey, T.: Mobile Robot Localization and Mapping in Extensive Outdoor Environments. PhD Thesis, Univ. of Sydney (2002)

[5] Davison, A.J., Murray, D.W.: Simultaneous localization and map-building using active vision. IEEE Tran. Pattern Analysis and Machine Intelligence 24(7), 865–880 (2002)

[6] Chatterjee, A., Matsuno, F.: A neuro-fuzzy assisted extended Kalman filter-based approach for Simultaneous Localization and Mapping (SLAM) problems. IEEE Trans. on Fuzzy Systems 15(5), 984–997 (2007)

[7] Chatterjee, A.: Differential evolution tuned fuzzy supervisor adapted extended Kalman filtering for SLAM problems in mobile robots. Robotica 27(3), 411–423 (2009)

[8] KOALA User Manual, Version 2.0 (silver edition), K-team S.A., Switzerland (2001)

[9] Nishimoto, T., Yamaguchi, J.: Three dimensional measurements using fisheye stereo vision. In: SICE Annual Conference, Japan, pp. 2008–2012 (September 2007)

[10] Brichfield, S.: KLT, An implementation of the Kanade-Lucas-Tomasi feature tracker, `http://www.ces.clemson.edu/~stb/klt`

[11] Shi, J., Tomasi, C.: Good Features to Track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994) Seattle, pp. 593–600 (June 1994)

[12] Marr, D., Poggio, T., Ullman, S.: Bandpass channels, zero-crossing, and early visual information processing. Journal of the Optical society of America 69, 914–916 (1979)

[13] Moravec, H.: Obstacle avoidance and navigation in the real world by a seeing robot rover. PhD thesis Stanford University (September 1980)

[14] Yamaguti, N., Oe, S., Terada, K.: A Method of distance measurement by using monocular camera. In: SICE Annual Conference, Japan, pp. 1255–1260 (July 1997)

[15] Chatterjee, A., Ray, O., Chatterjee, A., Rakshit, A.: Development of a Real-Life EKF based SLAM System for Mobile Robots employing Vision Sensing. Expert Systems with Applications 38(7), 8266–8274 (2011)

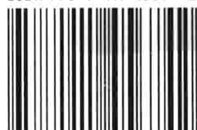
[16] Chatterjee, A., Singh, N.N., Ray, O., Chatterjee, A., Rakshit, A.: A two-camera based vision system for image feature identification, feature tracking and distance measurement by a mobile robot. International Journal of Intelligent Defence Support Systems 4(4), 351–367 (2011)



国际视野　科技前沿

国际信息工程先进技术译丛

《基于视觉的自主机器人导航》
《无线神经接口的超低功耗集成电路设计》
《基于片上去耦电容的配电网络》（原书第2版）
《智能摄像机》
《车载系统和安全的数字信号处理》
《嵌入式系统设计 —— 嵌入式信息物理系统基础》（原书第2版）
《纳米封装 —— 纳米技术与电子封装》
《内容分发网络》
《全面的功能验证：完整的工业流程》
《无线**Mesh**网络架构与协议》
《**UMTS**蜂窝系统的**QoS**与**QoE**管理》
《半导体制造与过程控制基础》
《**WCDWA**原理与开发设计》
《下一代移动系统：**3G/B3G**》
《**IMS:IP**多媒体概念和服务》（原书第2版）
《下一代无线系统与网络》
《深入浅出**UMTS**无线网络建模、规划与自动优化：理论与实践》
《**HSDPA/HSUPA**技术与系统设计——第三代移动
《通信系统宽带无线接入》
《无线传感器及元器件：网络、设计与应用》
《印制电路板——设计、制造、装配与测试》
《**IPTV**与网络视频：拓展广播电视的应用范围》
《多电压**CMOS**电路设计》
《微电子技术原理、设计与应用》
《蜂窝网络高级规划与优化**2G/2.5G/3G/…**向**4G**的演进》
《基于蜂窝系统的**IMS**——融合电信领域的**VoIP**演进》
《无线网络中的合作原理与应用》
《电生理学方法与仪器入门》
《移动电视：**DVB-H**、**DMB**、**3G**系统和富媒体应用》
《环境网络：支持下一代无线业务的多域协同网络》
《基于射频工程的**UMTS**空中接口设计与网络运行》
《未来**UMTS**的体系结构与业务平台：全**IP**的**3G CDMA**网络》
《**UMTS-HSDPA**系统的**TCP**性能》
《宽带无线通信中的空时编码》
《数字图像处理》（原书第4版）



上架指导　工业技术／电子技术／机器人

ISBN 978-7-111-45674-2　定价：59.90元